CONTEXT & SEMANTICS IN NEWS & WEB SEARCH

DAAN ODIJK

Context & Semantics in News & Web Search

Daan Odijk

Context & Semantics in News & Web Search

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof.dr. D.C. van den Boom
ten overstaan van een door het College voor Promoties ingestelde
commissie, in het openbaar te verdedigen in
de Aula der Universiteit
op vrijdag 10 juni 2016, te 13:00 uur

door

Daan Odijk

geboren te Velsen

Promotiecommissie

Promotor:		
	Prof. dr. M. de Rijke	Universiteit van Amsterdam
Co-promotor:		
	Dr. E.J. Meij	Yahoo Labs
Overige leden:		
	Prof. dr. F.M.G. de Jong	Universiteit Utrecht
	Prof. dr. B. Larsen	Aalborg University
	Dr. M.J. Marx	Universiteit van Amsterdam
	Dr. C.G.M. Snoek	Universiteit van Amsterdam
	Prof. dr. M. Worring	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

The research was supported by the Dutch national program COMMIT.

SIKS Dissertation Series No. 2016-20
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Cover by David P. Graus
Printed by Off Page, Amsterdam

ISBN: 978-94-6182-686-2

Acknowledgements

You are holding the fruits of nearly five years of my hard labour. I thoroughly enjoyed every minute of it, not in the least because of the wonderful people that have supported me and that I would like to acknowledge here.

First of all, I would like to thank my promotor Maarten. You have created the best possible environment for young researchers to grow and excel in. Thank you for this and for pushing me to achieve more, for giving me the freedom to go off the path and for being an inspiring mentor. Edgar, thank you for your all-round guidance (way before you became my co-promotor), your contagious enthusiasm, and for involving me in all sorts of schemes, often related to work, but probably more often and certainly more fun when not. Franciska, Birger, Maarten Marx, Cees and Marcel, thank you for serving as my committee members, taking the time to read my thesis and attending the defense ceremony.

I have spent a good part of the last fifteen years as a student and employee at the University of Amsterdam. I greatly appreciated the support, advice and guidance that I received over the years from Marcel, Maarten Marx, Christof, Evangelos and Cees. I thank the ILPS graduates that went before me for the many breadcrumbs that they have left along the way. Thank you, Edgar, Wouter, Katja, Marc, Bouke and Manos, also for the fun times we shared.

I learned a lot from my interactions with my fellow PhD students, both in ILPS and in COMMIT, and the students that I supervised. Some ILPSers have been alongside me for most of the journey and have quickly become good friends. I thank Hendrike for her creative bursts of energy, Richard for his contemplation, Ork for sharing a passion for all things visual, Anne—the co-author I never had—for caring to make every single thing better, David for his humor and unlimited enthusiasm, Marlies for being a good sport, and finally Tom, Ridho, Zhaochun, Nikos & Evgeny for our many fun discussions.

Many people have made my life as a PhD student a lot easier. I thank Petra & Caroline for the administrative support and the many fun chats, Isaac & Lars for good code and conversations, and the ladies of the COMMIT buro for running a smooth project. I thank SURFsara, FEIOG and Auke in particular for the great infrastructural support.

I thank all my co-authors for their valuable contributions to the work in this thesis and to my development as a scientist. Björn & Rens, thank you for a fruitful interdisciplinary collaboration. Cristina, Jacco, Kees, Laura & Thomas thanks for the collaboration in the QHP project.

While working towards my PhD, I had the opportunity to do two internships that have greatly enriched my PhD experience. It was an honor to work in the CLUES group at Microsoft Research and I thank Susan, Ryen & Ahmed for this opportunity, their support, and their patience when I threw yet another plot at them. I thank Bob, Mark, Petr & Clara for the fun we had in and around Seattle. Thanks to the TWCers, and Bart, Max & Rutger in particular, for seeing the products through the goofy science.

Ik prijs mij gelukkig met vrienden die voor mij klaar staan, hoe lang ik ze ook heb verwaarloosd. Bedankt, Julian, voor goede gesprekken, gegarandeerde ontspanning en bergen lol. Jurgo & Thijs, bedankt voor de broodnodige kwinkslagen.

Mijn ouders, Nardi & Marja, jullie onvoorwaardelijke steun, aanmoediging en geduld, hebben op een onbeschrijfelijke wijze een stempel gedrukt op dit proefschrift en wie ik ben geworden. Ons gezin is altijd als een warm bad geweest, waarin mijn nieuwsgierigheid alle ruimte kreeg en gevoed werd. Jasper & Pieter, bij de eerste promotieplechtigheid die ik bijwoonde wist ik al dat als ik daar ooit zou mogen staan, ik jullie, mijn broers, daar naast mij wilde hebben. Ik dank jullie en jullie partners Marleine & Stephanie voor precies de juiste hoeveelheid interesse in mijn wetenschappelijke leven en vooral voor uitstekende afleiding daarbuiten.

Tot slot, mijn lieve Fleur, wij ontmoetten elkaar als jonge promovendi, we hebben daarna iedere stap samen genomen en verdedigen nu in dezelfde week onze proefschriften. Jouw enthousiasme, intellect en geloof in mijn kunnen hebben mij een beter mens gemaakt.

Contents

1 Introduction

Information retrieval (IR) systems aim to provide users with easy access to information of their interest [12]. Searching the web via well-known web search engines, such as Baidu, Bing, and Google, has become one of the most prominent applications of information retrieval (IR) technologies, but as we will see in this thesis, there are many others. Scientific research in IR is often algorithmic in nature where the algorithms are meant to support effective access to information, with a strong emphasis on the evaluation of how well an IR system is able to fulfill the information need of a user. The scope of research in the area is broad, dealing not just with the many aspects of accessing information, but also with representation, storage and organization of information [12, 134]. In this thesis we touch on three of these aspects of IR research: the *domain* in which an IR system is used, the *users* interacting with the system, and the different access *scenarios* in which these users engage with an IR system.

The first aspect of interest for this thesis is the *domain* in which an IR system is used. For example, information retrieval in the web domain has specific challenges, such as the large volume of data, its diversity and fast pace of change [12]. Yet, in the domain of multimedia retrieval (e.g., image, audio or video search), challenges are of a very different nature, centered around issues such as representation, segmentation and matching of content. In news search, it is particularly challenging to understand the context of articles, as news is versatile, fluid, and background information is often important but limited. News stories are volatile and often presented without much context or with implicit context in case of a developing story, where it is typically assumed that news consumers are aware of current affairs.

A second general aspect of IR research of particular interest for this thesis concerns the *users* interacting with an IR system and their behavior. When a user interacts with an IR system, this is usually part of a larger information seeking task that takes place in a specific context [101], determined by a user's task and situation. Whether an IR system is dealing with a researcher studying a large collection of news articles, a leaned-back TV viewer seeking to be entertained or any user searching on the web, greatly influences the system and the support that a user needs. In recent years, there is an increased interest in user behavior in longer search sessions as opposed to one-off searches [60, 92, 166, 201]. In web search, such long search sessions are prevalent and time-consuming, e.g., around half of all Web search sessions contain multiple queries [201] and about 40% span three or more minutes [60]. While only 5% of these sessions last longer than 30 minutes, they account for about half of the time spent on web search. These longer sessions may involve exploring with learning [135, 228]. More frustrating for a user, however, is when no progress seems to be made at all and users are struggling to find what they are looking for. Hassan et al. [92] found that searchers' actions in long sessions suggested that they were exploring in 40% of the sessions and struggling in 36%, while showing behavior of both in the remaining 24%. Whether the user is struggling or exploring, understanding the specific context in which they seek is essential to help fulfill their information need.

A third important aspect of IR research that is relevant to this thesis is the *scenario* of how a user accesses an IR system. In the classical IR model [12], information is stored in documents that are organized in a collection, and it is being accessed by a user with an information need [101, 134]. IR systems typically deal with such information needs, as conveyed in a query with a few descriptive terms, by returning a ranked list of documents [134]. Next to this so-called ad-hoc search scenario, new interaction scenarios are emerging, as IR systems are becoming more wide-spread. New search scenarios may be more conversational when using a personal digital assistant, such as Apple's Siri, Google Now and Microsoft's Cortana. Instead of a user visiting a search engine's webpage and typing in a query, search is more and more in situ. Increasingly, these IR systems are able to leverage insights about the context of an information need beyond or even before an expressed query. In such a pro-active search scenario, the context may be given by a user's location [22] or activities [60, 64, 128]. These new interaction scenarios signal a move from pulling to pushing information, accelerated by constant access to the web on smart phones and tablets. An increasingly familiar example concerns the exploratory search scenario, where a user does not know beforehand what they are looking for, nor where to find it [135].

This thesis provides contributions on each of the three aspects listed above: *domains*, *users* and *scenarios*. It is organized around three research themes: (1) studying news collections, (2) struggling and success in web search, and (3) pro-active search for live TV. In each of the research themes, the domain, user and scenario aspects figure prominently, as we will explain below.

1.1 Research Outline, Themes and Questions

As pointed out above, three research themes guide the research presented in this thesis, touching on three aspects of IR research: the *domain* in which an IR system is used, the *users* interacting with the system, and the different access *scenarios* in which these users engage with an IR system. Table 1.1 lists the research themes and highlights how these three aspects of IR research figure in that theme. Central to these research themes is the aim to gain insights and develop algorithms that support searchers in their quest, whether it is a researcher exploring or studying a large collection, a web searcher struggling to find something or a television viewer searching for related content. The algorithms and insights presented here aim to ease some of the "pains" that searchers experience while using an IR application. For each research theme, we look at the behavior of a user in a specific domain and scenario and propose new algorithms to provide users with easy access to information. The research themes and questions are outlined in more detail in this section.

Table 1.1: The three research themes covered in this thesis, listing for three aspects of IR research how they figure in that research theme.

Research theme	Domain	User	Scenario
1. Studying news collections	News	Researcher	Studying
2. Struggling and success in web search	Web	Struggling user	Long session
3. Pro-active search for live TV	Media	TV viewer	Pro-active

Theme 1—Studying News Collections

In the first research theme, the users that we focus on are researchers who study large collections of news articles. Inspired by information seeking tasks from the social sciences and humanities, we propose new IR algorithms. We start our investigation by focusing on how researchers study and explore large collections. A huge amount of digital material has become available to study our recent history, ranging from digitized newspapers and books to, more recently, encyclopedic web pages about people, places and events. With that, the wish grows for researchers to explore and study these collections and in particular to study the digitized news articles in their context.

Looking at how historians collect relevant material, Duff and Johnson [59] identified four types of information seeking tasks that they employ: (1) orientation, (2) known material search, (3) building contextual knowledge and (4) identification of relevant material. The first task orientation, relates closely to the IR concept of exploratory search [135], where a user does not know beforehand what they are looking for, nor where to find it. When exploring a large collection, information aggregated over documents can help maintain a sense of context [234]. As a motivational use case, we present an exploratory search interface for a large news collection in which we use visualizations of the entire collection and of specific parts of the collection. We investigate how researchers from the humanities select subsets of large collections for close examining. This document selection process is a combination of the first, second and fourth task of Duff and Johnson [59] and we describe how exploratory search techniques fit within these tasks.

We study in more detail the information seeking task of building contextual knowledge to ground further research [59]. Humanities scholars critically regard historical sources in their context considering aspects such as date, authorship and localization in order to assess the credibility of a source [78]. When multiple sources are considered, each might provide an interestingly different perspective on a historical events, expressing views at the time of writing or even a subjective view of the author. Where news articles have a clear temporal footprint, other sources such as encyclopedic articles might not. For these, temporal references can be extracted and leveraged in combination with the textual content to find related articles. We cast this as an IR task and ask:

RQ1 Can we effectively extract temporal references from digitized and digital collection and does leveraging these temporal references improve the effectiveness of retrieving related articles?

To illustrate how the algorithms developed to answer RQ1 can be used, we present an exploratory search interface that highlights different perspectives and describe a use case on how a historian can use this to explore and analyze the connected collections.

Next, we turn our attention to the social sciences, where mass communication (such as news) is often studied through a methodology called *content analysis* and evolves around the question: “Who says what, to whom, why, to what extent and with what effect?” [121]. We study in more detail how news stories are presented. In communication science, framing is the way in which journalists depict an issue in terms of a ‘central organizing idea’ [76]. Frames can be seen as a perspective on an issue. Complex characteristics of messages such as frames have been studied using thematic content analysis [202]. Indicator questions are formulated, which are then manually coded by humans after reading a text and combined into a characterization of the message. To scale frame analysis up to large collections, we operationalize this as a classification task and ask the following research question:

RQ2 Can we approach human performance on the task of frame detection in newspaper articles by following the way-of-working of media analysts?

Following the way-of-working of media analysts, we propose a two-stage approach, where we first rate a news article using indicator questions for a frame and then use the outcomes to predict whether a frame is present.

Theme 2—Struggling and Success in Web Search

In the second research theme of this thesis, we turn our attention to the domain of web search and a long session scenario. Web searchers sometimes struggle to find relevant information. Struggling leads to frustrating and dissatisfying search experiences, even if searchers ultimately meet their search objectives [92]. A better understanding of search tasks where people struggle is important in improving search systems. Key components of support for struggling users are algorithms that accurately predict the nature of future actions and their anticipated impact on search outcomes.

When searchers experience difficulty in finding information their struggle may be apparent in search behaviors such as issuing numerous search queries or visiting many results within a search session [11]. Methods have recently been developed to distinguish between struggling and exploring in long sessions using only behavioral signals [92]. However, little attention has been paid to *how* and *why* searchers struggle. This is particularly important since struggling is prevalent in long tasks, e.g., Hassan et al. [92] found that actions in 60% of the long sessions that they studied suggested searchers were struggling.

We address this important issue using a mixed methods study using large-scale logs, crowd-sourced labeling, and predictive modeling and ask two research questions:

RQ3 How do web searchers behave when they cannot find what they are looking for?

RQ4 How do web searchers go from struggle to success and how can we help them make this transition?

Among other things, our results show that less successful struggling searchers use more spelling corrections, substitutions, completely new queries and returning to previous queries. It appears that less successful searchers experience more difficulty selecting the

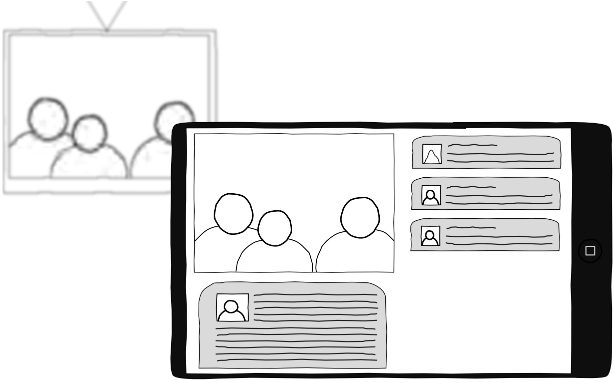


Figure 1.1: Sketch of a second screen device (e.g., a tablet or smartphone) showing links to additional background information, synchronized with a live television broadcast. In this sketch, the background information is displayed as information cards, showing a list of cards on the right and a highlighted card on the bottom left.

correct query vocabulary. Motivated by this, we continue the research in this thesis in a pro-active search setting, where we try to automatically generate queries and find relevant content for a user, based on the context of their search.

Theme 3—Pro-active Search for Live TV

In the third and final research theme in this thesis, we consider a pro-active search scenario, specifically in a live television setting, where we propose algorithms that leverage contextual information to retrieve diverse related content for a leaned-back TV viewer.

The way people watch television is changing [79, 156, 180]. Increasingly, broadcasts are consumed interactively and with additional content related to what they are watching. Around 70% of tablet and smartphone owners use their devices while watching television [155]. Razorfish [180] found that 38% of "mobile multitaskers" access content that is related to the TV program they are watching. This allows broadcasters to provide consumers with additional background information that they may consume right away or bookmark for later consumption. With the rich context of a live TV program and users seeking related content, there is a clear need for IR algorithms and applications that can leverage such contextual information. Figure 1.1 shows a sketch of an application that allows viewers to access additional background content. In the third research theme, such additional background information is automatically provided in real time, by analyzing the subtitles that come with the television broadcast.

In a live TV setting, the main challenge is to find content while a story is developing. For live TV, subtitles are often available (provided for the hearing impaired). Using this textual stream of subtitles, we can automatically provide links to background information and generate search queries. This task has unique demands that require an approach that needs to (1) be high-precision oriented, (2) perform in real time, (3) work in a streaming setting, and (4) typically, with a very limited context. We therefore ask the following two research questions:

RQ5 Can we effectively and efficiently provide background information for a live television broadcast in real time using an entity linking approach and does explicitly modeling streaming context improve the effectiveness?

RQ6 Can we effectively and efficiently find video content related to a live television broadcast in real time using a query modeling approach?

The research chapters of this thesis (Chapters 3–5, 6 and 7) describe the work on answering the six research questions listed above. Answers to the research questions are provided at the end of each chapter. Our findings are summarized in a final concluding chapter (Chapter 8). In the next sections we list the contributions of this thesis and provide an overview of the thesis and its origins.

1.2 Main Contributions

In this section, we summarize the main theoretical, algorithmic and empirical contributions of this thesis.

Theoretical contributions

- **Exploring Historical Perspectives** – We describe how researchers from the humanities can use automatically created connections between digital collections to explore and analyze perspectives in a historical collections.
- **Characterization of Struggling Behavior** – We characterize the behavior of users that are struggling to find what they are looking for and describe differences in behavior given task success.
- **Real-Time Entity Linking** – We formalize the task of real-time entity linking of a textual stream.
- **Related Content Finding as an MDP** – We formalize the task of related content finding to a live television broadcast as a Markov decision process.

Algorithmic contributions

- **Connecting Digital Collections** – We propose an approach to connect digital collections through automatically extracted temporal reference and textual content.
- **Frame Detection in News** – We propose a two-stage approach to finding frames in news articles. We start by predicting the outcomes to indicator questions associated with a frame and then use the predicted outcomes to decide about the presence of the frame in a given text.
- **Reformulation Strategy Classifier** – We develop a classifier to predict query reformulation strategies during struggling search tasks. We provide a method that can accurately classify query reformulations according to an intent-based schema that can help select among different system actions. Additionally, we provide a method that can accurately identify pivotal (turning point) queries within search tasks in which searchers are struggling.

- **Real-Time Entity Linking** – We propose a set of effective feature-based methods for performing real-time entity linking. We explore multiple link generation and link pruning approaches and thoroughly evaluate the effects on both efficiency and effectiveness. We extend this approach with a graph-based and a retrieval-based method to keep track of context in a textual stream.

- **Dynamic Query Modeling** – We propose a dynamic query modeling approach, using reinforcement learning to optimize a retrieval model that is sufficiently efficient to be run in near real time in a live television setting and that significantly improves retrieval effectiveness over state-of-the-art baselines for stationary query modeling and for text-based retrieval in a television setting.

Empirical contributions

- **Extracting Temporal References** – We propose a method that extracts temporal references with satisfactory accuracy and show that we can use these references for the task of related article finding. We identify interesting challenges in extracting temporal references from historical narratives and digitized text.

- **Frame Detection in News** – We show how explicitly modeling the manual thematic content analysis approach does not improve performance on the task of frame detection in news.

- **Struggling and Success** – We use large-scale search log analysis to characterize aspects of struggling search tasks and to understand how some tasks result in success, while others result in failure.

- **From Struggling to Success** – We propose and apply a crowd-sourced labeling methodology to better understand the nature of the struggling process (beyond the behavioral signals present in log data), focusing on why searchers struggled and where it became clear that their search task would succeed (i.e., the pivotal query).

- **Real-Time Entity Linking** – We show how a learning to rerank approach for entity linking performs on the task of real-time entity linking, in terms of effectiveness and efficiency. We show how explicitly modeling context can further improve effectiveness. By investigating the effectiveness and efficiency of individual features we provide insight in how to improve effectiveness while maintaining efficiency for this task.

- **Dynamic Query Modeling** – We provide a thorough evaluation and an analysis of when and why our dynamic query modeling approach for related content finding for live TV works. We find that adding more weighted query terms and decaying term weights based on their recency significantly improve the effectiveness.

In addition to the contributions listed above, this thesis provides contributions in the form of resources, software and demonstrators.

Resources

- **Temporal References Extracted from the Books of Loe de Jong** – We contribute automatically extracted temporal references in the books of Loe de Jong as Linked Open Data.
- **Entity Linking for DWDD** – We create a dataset for the task of entity linking on a textual stream, including ground truth. The dataset consists of more than 1,500 manually annotated links in over 5,000 lines of subtitle for 50 video segments.
- **NOS Journaal Related Videos** – We create a dataset for the task of related content finding to a live television broadcast. For 50 news video segments from May 2013, we annotate on average 79 archival videos per segment for relevance on a five-point scale.

Software and Demonstrators

- **Quantifying Historical Perspectives** – A novel search interface that supports researchers from the humanities in studying different perspectives on WWII.
- **ShoShin** – An exploratory search interface that guides the user to interesting periods in time and interesting bits of information, leveraging the fact that the users are experts on the topic. Furthermore, we build upon ShoShin to provide a coordinated environment for understanding the development of word usage and meaning through time.
- **ThemeStreams** – An interactive visualization aimed at giving insight into the ownership and dynamics of themes being discussed, thereby enabling users to answer questions such as *Who put this issue on the map*?
- **Semanticizer** – Our approach for real-time entity linking is publicly available as a webservice and released as open-source software.

1.3 Thesis Overview

The research chapters in this thesis are organized in two parts. Part I covers research theme 1 in Chapters 3 and 4, whereas Part II covers both research themes 2 and 3. Research theme 2 (Chapter 5) motivates the investigations within research theme 3 (Chapters 6 and 7). Figure 1.2 shows a graphical overview of the contents of this thesis. Part I and II of this thesis can be read independently. Chapter 2 provides the necessary background for both parts. Readers familiar with IR could skip this chapter. Chapter 4 can be read without this background. Chapter 5 of Part II provides the motivation for the subsequent chapters and can be read independently.

Chapter 2 provides the background for this thesis and is organized along the three research themes, after a first broad introduction of IR. After this introduction, a reader interested only in a particular research theme can continue to the first section and the relevant section of Chapter 2, the research chapters of that theme, and then finish with the relevant parts of Chapter 8. This section gives an overview of the content for both parts and for each chapter of this thesis.

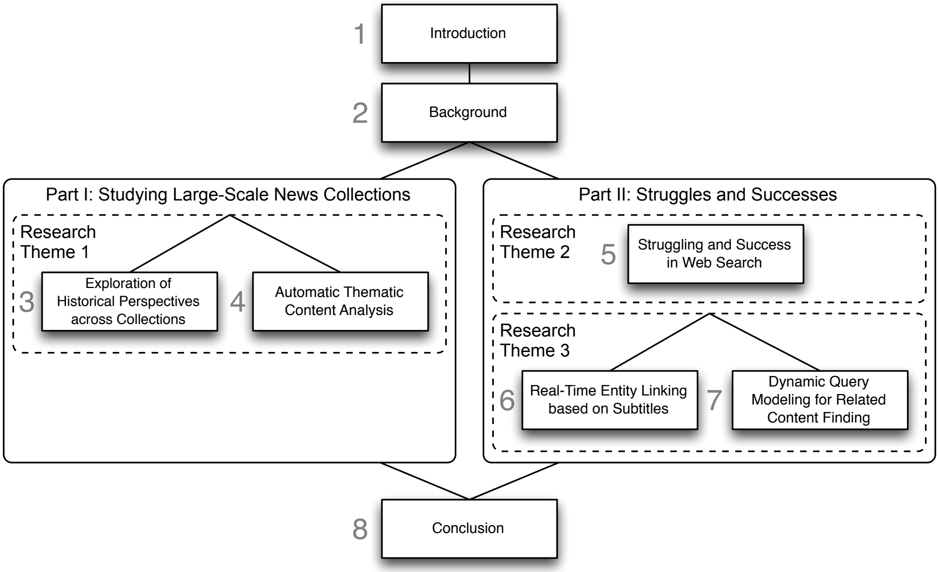

1
Introduction
2
Background
Part I: Studying Large-Scale News Collections
Research Theme 1
3
Exploration of Historical Perspectives across Collections
4
Automatic Thematic Content Analysis
Part II: Struggles and Successes
Research Theme 2
5
Struggling and Success in Web Search
Research Theme 3
6
Real-Time Entity Linking based on Subtitles
7
Dynamic Query Modeling for Related Content Finding
8
Conclusion

Figure 1.2: Overview of the structure of this thesis that consists of two parts and covers three research themes. Each part and each research theme can be read separately.

Chapter 2—Background

This chapter introduces background for the subsequent chapters in this thesis. Related work is covered along the three research themes, preceded by a broad introduction of the field of IR.

Part I—Studying Large-Scale News Collections

The first part of the thesis opens with a study of the ways in which researchers study and explore large-scale news collections. Motivated by how humanity scholars select documents for close reading, this part introduces motivational exploratory search interfaces. We also include two more algorithmic contributions: (1) novel methods for connecting collections, and (2) an automatic thematic content analysis approach based on how communication scientists study framing in news.

Chapter 3—Exploration of Historical Perspectives across Collections

We present a motivational use case on how historians use exploratory search to interactively select relevant documents from very large news collections. In this chapter, we propose to connect heterogeneous digital collections through temporal references in documents and their textual content. We evaluate our approach and find that it works very well on digital-native collections. Digitized collections pose interesting challenges that we address with additional preprocessing. We illustrate the use of these algorithms in a use case with a novel search interface to explore and analyze the connected collections, highlighting different perspectives.

Chapter 4—Finding Frames in News

Based on how communication scientists study framing in news, in this chapter, we propose an automatic thematic content analysis approach. Framing in news is the way in which journalists depict an issue in terms of a 'central organizing idea.' We explore the automatic classification of four generic news frames: conflict, human interest, economic consequences, and morality.

Part II—Struggles and Successes

The second part of the thesis is motivated by a mixed-methods study on how web searchers behave when they cannot find what they are looking for. In this part, two methods to automatically retrieve content based on subtitles are introduced, one using entity linking, and one that uses reinforcement learning to generate effective queries for finding related content. Both methods are highly efficient and are currently used in a live television setting in near real time.

Chapter 5—Struggling and Success in Web Search

Part two of the thesis is motivated by a mixed-methods study on how web searchers behave when they cannot find what they are looking for. This so-called struggling leads to frustrating and dissatisfying search experiences, even if searchers ultimately meet their search objectives. Based on large-scale log analysis, crowd-sourced labeling and predictive modeling, recommendations are made for how search systems can help users struggle less and succeed more.

To ease some of the search "pains" identified in Chapter 5, novel methods for automatically searching for background information are proposed in Chapters 6 and 7. These methods are evaluated in a live television setting.

Chapter 6—Real-Time Entity Linking based on Subtitles

In this chapter, we consider the task of linking textual streams derived from live broadcasts to Wikipedia. This allows broadcasters to provide consumers with additional background information that they may bookmark for later consumption. We propose an effective learning to rerank approach whose processing time is very short. We extend this approach, leveraging the streaming nature of the textual sources that we link by modeling context explicitly.

Chapter 7—Dynamic Query Modeling for Related Content Finding

In this chapter, we consider the task of finding video content related to a live television broadcast for which we leverage the textual stream of subtitles associated with the broadcast. We propose a method that uses reinforcement learning to directly optimize the retrieval effectiveness of queries generated from the stream of subtitles. Our method is highly efficient and can be used in a live television setting, i.e., in near real time.

The final chapter in this thesis returns to the research questions from the introduction and looks ahead.

Chapter 8—Conclusions

In this chapter, we draw conclusions and provide an outlook on future research.

1.4 Origins

This thesis is based on ten publications in total [38, 57, 159–162, 164–167]. For each research chapter, we list the publications on which it is based, specifying the role of each co-author for each publication.

Chapter 3 is primarily based on *Supporting Exploration of Historical Perspectives across Collections* published at TPDL'15 by Odijk, Gârbacea, Schoegje, Hollink, Boer, Ribbens, and Ossenbruggen [164]. Gârbacea and Odijk designed the algorithms. Gârbacea, Odijk and Schoegje designed and implemented the application. Odijk initiated the writing and all authors contributed. The motivational use case presented in the beginning of the chapter is based on *Semantic Document Selection: Historical Research on Collections that Span Multiple Centuries*, published at TPDL'12 by Odijk, de Rooij, Peetz, Pieters, de Rijke, and Snelders [159]. Odijk designed and implemented the interactive search application, with contributions from De Rooij and Peetz. Snelders and Pieters provided use cases. All authors contributed to the text.

Chapter 4 is based on *Automatic Thematic Content Analysis: Finding Frames in News* published at SocInfo'13 by Odijk, Burscher, Vliegenthart, and de Rijke [161] and *Teaching the Computer to Code Frames in News* published in Communication Methods and Measures by Burscher, Odijk, Vliegenthart, de Rijke, and de Vreese [38]. Odijk performed the experiments and designed the algorithms with contributions from Burscher and De Rijke. Odijk and De Rijke initiated the writing for SocInfo'13 [161] and all authors contributed to the text. Burscher performed similar and additional experiments, leading to the same outcomes. Burscher initiated writing for a difference target audience [38], framing the outcomes for an audience of communication scientists. All authors contributed to this text as well.

Chapter 5 is based on *Struggling and Success in Web Search* published at CIKM'15 by Odijk, White, Hassan Awadallah, and Dumais [166]. Odijk performed the log analysis, crowd-sourcing and prediction experiments. Odijk wrote a first version and all authors contributed to the text.

Chapter 6 is based on *Feeding the Second Screen: Semantic Linking based on Subtitles* published at OAIR'13 by Odijk, Meij, and de Rijke [162] and an extension under review for publication in a journal as *Real-Time Entity Linking based on Subtitles* by Odijk, Meij, and de Rijke [167]. Odijk led the development of the algorithms; the co-authors contributed. For both, Odijk performed the experiments and all authors contributed to the text.

Chapter 7 is based on *Dynamic Query Modeling for Related Content Finding* published at SIGIR'15 by Odijk, Meij, Sijaranamual, and de Rijke [165]. Odijk led the development of the algorithms; the co-authors contributed. Odijk performed the experiments with contributions from Sijaranamual. All authors contributed to the text.

The background on exploratory search in Section 2.2 shares its origin with Chapter 3 [164] and three demonstrator publications [57, 159, 160]. The background in Section 2.4 shares its origins with Chapters 6 and 7 [162, 165, 167] and uses material from tutorials on entity linking and retrieval presented at WWW'13, SIGIR'13 [142] and WSDM'14 [143].

Indirectly, the thesis also builds on joint work on exploring entity associations [181] and personalized historical events [80], and entity linking on multilingual video lectures [163] and topical social streams [221]. Finally, the thesis draws from insights and experiences gained by participating in evaluation campaigns [23, 77, 81, 203, 213].

2 Background

In this chapter, we discuss the background for the research presented in this thesis. We first introduce general concepts and terminology from information retrieval (IR) in Section 2.1, providing the shared background for all research presented in this thesis. We build on this background with relevant concepts and previous work specific to each of the three research themes presented above. In Section 2.2, we discuss work related to research theme 1, on studying news collections, exploratory search and connecting collections. Section 2.3 discusses background on research theme 2 covering understanding user behavior in web search. Section 2.4 concludes this chapter with background for research theme 3 on search in a streaming setting.

2.1 Information Retrieval

One of the first definitions of the scientific field of information retrieval (IR) comes from the 1968 textbook of Salton [187] and is still surprisingly accurate:

> Information retrieval is a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information.

As Salton's definition indicates, research in IR covers a broad scope, dealing not just with access to information, but also with representation, storage and organization thereof [12, 134]. Textual information has been the primary focus of IR research, but increasingly, applications of IR involve multimodal information. More and more, the content that is sought has structure and combines significant text content with other media [54]. Baeza-Yates and Ribeiro-Neto [12] describe the primary aim of IR systems as to provide users with easy access to information of their interest. An IR system is often called a search engine;[1] throughout this thesis we will use both terms.

In the classical IR model [12], information is stored in documents that are organized in a collection, and it is being accessed by a user with an information need [101, 134]. Users convey their information needs through a few descriptive terms in a query. IR systems deal with this query by returning a ranked list of documents [134]. Documents are commonly somewhat structured, with content organized in textual fields such as title

[1] Originally, the term "search engine" referred to specialized hardware for search [54]. Over the last 30 years, the term gradually came to be used as a synonym for, and even preferred over "information retrieval system".

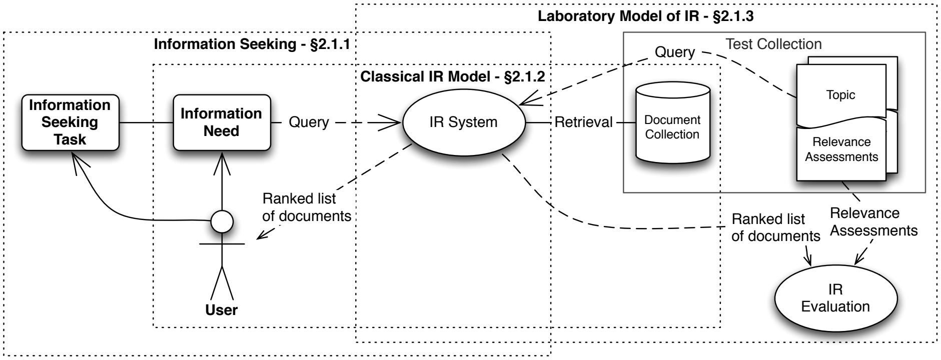


Figure 2.1: A graphical representation of the IR ecosystem, highlighting three predominant models of IR, each with a distinct emphasis.

or description. Additionally, metadata can be associated with the document, such as the author or creation date. In contrast to a record in a database, a (field in a) document in IR is represented as unstructured information, typically as textual content [54]. For retrieval, terms in documents may be stored in an *inverted index*, a data structure that maps terms to documents to improve the speed of search [54].

In the remainder of this section, we first discuss a broader view of how IR systems are used (§2.1.1), followed by fundamental models and tasks (§2.1.2) before discussing how they are evaluated (§2.1.3). Figure 2.1 shows a graphical representation of the IR ecosystem, highlighting three different views on information retrieval and how they relate to each other. We discuss the elements of this ecosystem in detail in the following sections.

2.1.1 Information Seeking & Interactive Information Retrieval

When a user interacts with an IR system, this is usually part of a larger information seeking task that takes place in a specific context [101], determined by a user's task and situation. In all research chapters in this thesis, we pay particular attention to the context of the IR task being studied and to how users interact with the IR system. We do this by motivating the research with use cases or large-scale data analysis and by describing how the research is applied. This attention to the context of IR tasks and the zoomed-out view on information seeking tasks touches on the field of information science [101, 205]. This field is closely related to IR, and it is broad and interdisciplinary, primarily concerned with the analysis, collection, classification, manipulation, storage, retrieval, movement, dissemination, and protection of information [205]. Both fields emerged around the same time, inspired by the post-war vision of Bush [39] of a *memex*:

> A memex is a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility.

Within the information science field, researchers in information behavior look at human behavior in dealing with generation, communication, use and other activities concerned with information [101]. We refer the reader to the survey of Case [46] for a recent review of information behavior research.

Relevant for the work presented in this thesis is the emphasis within the field of information science on the interactions of the user with an IR system. Ingwersen and Järvelin [101] describe an integrated view of information seeking and retrieval, with interaction as a bridging concept. Interactive IR involves the interactive communication processes [101] that occur during retrieval of information between participants in information seeking and retrieval, and, most importantly, between the searcher (in their context) and the search interface. The classical IR model considers the task of retrieving a ranked list of documents for a single query in isolation. In practice, queries are often part of a sequence of search interactions, called a "session." Jansen [103] defines the duration of a session as the interval between the first query and the user "leaving" the search engine. For practical purposes, a session timeout is typically used, e.g., when a user is inactive for 30 minutes [58, 226]. We use a similar operationalization for session in the research in Chapter 5.

2.1.2 Retrieval Models

To retrieve a ranked list of relevant results, IR researchers develop retrieval models and evaluate their effectiveness (as described below). A retrieval model represents the matching process of query and document. It produces a ranking of documents that match the query. In early IR systems, this matching process was set-based, using boolean queries [190] that allowed a user to express logical clauses for matching. For example, a query could be formulated to match *information AND retrieval OR seeking*.

An early extension to the boolean model was to allow users to weight query terms [189]. This way, the retrieval model ranks documents by their relevance to the query, starting with the most relevant document. A formalization of this is called the *vector space model* [188]. In this model, the query and documents are represented as vectors in a space, where each term is a dimension. Matching is done by comparing the distance between document and query in this space, e.g., by measuring cosine similarity. Assigning a weight to each term in this vector space has the effect of weighing the term in the document ranking. Commonly used weighting schemes use the frequency of a term in the document (term frequency or TF) combined with the number of documents that contain the term (document frequency or DF). The intuition behind the former is that if a query term occurs frequently in a document, that document is likely to better match the query. The intuition behind the latter is that words that occur in many documents are likely to be very generic and thus less useful for discriminating between relevant and non-relevant documents.

The concept of document ranking was further formalized by Robertson [185] as the *probability ranking principle*, which states that an optimal ranking is one that ranks documents in decreasing order of their probability of relevance to the query. The two most prominent retrieval models that have followed from this principle are: BM25 and the language modeling framework. The BM25 retrieval model [184] heuristically combines the TF and DF statistics described above and uses document length normalization. In the language modeling framework [175] documents are modeled as a bag of words drawn from a statistical language distribution. Computing a score for a document boils down to computing the probability that both the query and the document where drawn from the same distribution.

Learning to rank. An alternative approach to ranking documents is to apply machine learning in a so-called *learning to rank* setting [73, 130]. Here, ranking functions such as BM25 and statistics like as TF and DF are used as features in a machine learning approach to optimally rank documents. A document collection with relevance assessments for each query is used to learn a model that optimally combines these features into a single ranking score. In practice, many different kinds of features are considered, including for example: document length, readability, or features derived from link structure in web pages. A well known example of this last type is Pagerank [169], an algorithm that iteratively computes the influence of webpages based on the pages that link to it. In the research within the third research theme (Chapters 6 and 7), we use learning to rank to improve the quality of our search results.

In Chapter 7, we use an *online learning to rank* approach. In an online setting, ranking is optimized directly from user feedback [233] instead of from annotations in the offline setting. This online setting maps directly to approaches from reinforcement learning (RL) [207]. RL intertwines the concepts of optimal control and learning by trial and error. Central is the concept of an "agent" optimizing its actions by interacting with the environment. This is achieved by learning a *policy* that maps *states* to *actions*. In modeling the IR setting, Hofmann et al. [97] maps the retrieval system to role of *agent*, taking the *action* of retrieving documents given the *state* of a query on the basis of a *policy*.

To formalize the task of finding video content related to a live television broadcast in Chapter 7, we model it as a Markov decision process (MDP) [66]. An MDP is a specific type of reinforcement learning problem that was proposed before the field was known as reinforcement learning. In an MDP, we decide on the optimal action in a Markov process [17]. The Markov property holds when the policy for a state is independent on the previous states. A Markov state thus has to represent the entire history of previous states as far as this is relevant for the value of the policy. MDPs have been used to model diverse IR problems, for example, Jin et al. [108] utilize reinforcement learning and MDPs to improve ranking over multiple search result pages. Closer to our work in Chapter 7 is recent work to explicitly model user behavior in session search [83, 132, 133]. Guan et al. [83] decrease weights of new terms based on past rewards, whereas Luo et al. [132] model session search as a dual-agent stochastic game. Investigating the design choices for MDPs, they find that explicit feedback and selectively enabling or disabling specific retrieval technology are most effective in session search [133]. In our streaming setting, new information keeps coming in, hence we are dealing with a non-stationary MDP. More specifically, because the decision of what action to choose does not influence the states that emerge from the environment, this is considered an associative search task [15].

Beyond document retrieval. Most of what we have described above relates to the core IR task of *ad hoc search*, where a user poses any possible query and an IR system responds with a ranked list of documents. Many other IR tasks exists. We cover two that are particularly relevant to this thesis: *text classification* and *document filtering*.

In a *text classification* task, the aim is to assign a particular label to a document, or viewed differently, to assign a document to a particular class or category [12]. The goal of this task is to organize a collection of documents and allow better understanding and interpretation of the collection. An early example of text classification is librarians assigning topical labels to books (going back to the first library around 300 BC [172]). One of the most prominent current examples is that of spam detection, e.g., in email or in web search [54]. We consider a diverse set of text classification tasks in this thesis, classifying temporal references (Chapter 3), frames in news (Chapter 4), query reformulation strategies (Chapter 5) and links to background information (Chapter 6). The algorithms for this task closely resemble those used a retrieval setting. We employ an approach similar to the learning to rank approach described above, in which we use features derived from textual content to learn a model that assigns labels as accurately as possible. We describe our methods in detail in the research chapters.

Another common IR task is *document filtering*. In a typical search task the query changes and the collection remains static. In a typical document filtering task, a standing query is used to filter a stream of documents [16]. No ranking of documents is necessary. The work we present in Chapter 6 combines the ad hoc search and document filtering tasks in searching for background information based on a textual stream. Other examples of such tasks include summarizing social media in real time [182] and finding replications of news articles while they appear [212].

Temporal IR. The models we propose in Chapters 3 and 7 leverage temporal information in some form for finding related content. In Chapter 3, we extract temporal references and combine a notion of temporal similarity with textual similarity to find related documents. In Chapter 7, we use the recency of a term in a textual stream as an indication for how useful that term is as a query term to find related content. Temporal IR deals with modeling temporal patterns to improve information retrieval and covers topics such as document freshness and temporal relevance. Alonso et al. [7] review current research trends in temporal IR and identify open problems and applications. Open problems include how to compute temporal similarity, how to combine scores for textual and temporal queries, and presenting temporal information in an interactive setting. Related to our work in Chapter 7, Efron [61] uses linear time series models for weighting terms, where the time series are computed on the target document collections. Similarly, Peetz et al. [174] use temporal bursts in a microblog collection to reweight query terms for improved retrieval. Our work differs in that we model temporal dynamics in the stream of subtitles from which we generate queries.

2.1.3 Information Retrieval Evaluation

A core issue in IR research is the evaluation of IR systems. As a scientific discipline, IR has a strong experimental tradition that is based on what is sometimes called the *laboratory model of IR* [101] (depicted on the right in Figure 2.1). In this evaluation paradigm, a test collection is created by formulating a set of queries (called topics) for which the relevance of documents from a reference collection is assessed by judges. IR system are then evaluated on their ability to find the relevant documents from the reference collection. This prototypical evaluation model has its roots in the Cranfield studies in the 1950s and 1960s [50] and was adopted by the TREC evaluation initiative [219], which gave a big boost to research on IR in the early 1990s [208, 216]. Early IR test collections focused on scientific documents [50, 71, 217], with more emphasis on news articles in the TREC initiative (about half of the test collections used for the first eight TREC evaluation campaigns consisted of news articles [220]).

The experiments presented in this thesis follow the set-up common in IR (the laboratory model [101] or Cranfield paradigm [50]). We design contrastive experiments that compare novel IR systems (or variants of systems) to a so-called baseline system. This comparison is done using evaluation metrics (detailed below), computed over topics in a test collection. Performance differences from the baseline system are tested for significance using a statistical test, such as a paired t-test by pairing the computed metric per topic for two systems. For more details on the history of this evaluation methodology for evaluating systems with relevance assessments, we refer to recent IR textbooks [12, 40, 54, 134].

An interesting alternative to this paradigm is to evaluate an IR system directly with users. Kelly [112] comprehensively summarizes different methods for evaluating search systems with searchers. For the research presented in the first and the third research theme, we did not have direct access to large numbers of users, nor their feedback. In the second research theme of this thesis, we look at the behavior and success of users that cannot find what they are looking for. We describe the related background for this in more detail in Section 2.3.

Evaluation metrics. To evaluate the results produced by an IR system according to the laboratory model, relevance assessments for a topic from the test collection are used to compute an evaluation metric. This metric is computed over a ranked list of documents produced by the system for the query associated with the topic. Per topic, the item-based relevance assessments are aggregated to a single result-based score. Besides the experimental procedures described above, the early work of Cleverdon [50] also introduced the first evaluation metrics to compare the results produced by IR systems. He introduced two intuitive evaluation metrics that are still popular: precision and recall. Precision is the proportion of the retrieved documents that are relevant. Recall is the proportion of relevant documents that are retrieved. The assumption here is that all relevant documents for a given query are known, which is less and less the case in current test collections, as these become bigger (e.g., the Clueweb12 test collection consists of 733 million web pages [42]).

To remedy this and other shortcomings, a wide palette of evaluation metrics and variants have been proposed [12, 104, 134, 214]. For example, the F-measure [214] combines precision and recall into a single value. Another single value metric that is commonly used is precision at a specific cutoff (P@k); this metric is appropriate in settings where users typically do not inspect retrieved documents beyond the first result page, such as in web search. The cutoff point k should be motivated by how the IR system that is evaluated will be used. In the R-precision metric, this threshold is set to the number of relevant documents for a given query. The threshold k will be different per query if the number of relevant documents differs. This makes it easier to interpret the scores for R-precision when averaged across queries [134]. The mean average precision (MAP) metric has similar characteristics. Per query, the average is computed of the precision at the position of each relevant document in the ranking. The final metric is the mean over queries of these average precision values.

Recently proposed metrics put more emphasis on how a user interacts with the search results, for example by modeling clicks from historical results [48]. One of the most important issues that these metrics address is the so-called position bias [110], i.e., users are more likely to inspect results higher in the ranked list. Carterette [44] showed that many metrics (e.g., [104, 150]) fit in a conceptual framework of utility-based metrics. This combines a notion of the usefulness or utility of a document, with a discount function based on the document's rank [44]. The metrics boil down to the same basic formula, composed of a sum of the product of the document utility and the discount factor:

$$M = \sum_{k=1}^{K} gain(rel_k) \times discount(k) \tag{2.1}$$

In Chapters 3 and 7, we use a metric from this family called discounted cumulative gain (DCG) [104]. DCG models the utility of a document using a gain function that based on relevance assessments. The gain function in DCG can simply be the value of the relevance annotation: rel_k, which is usually between -2, and 4, where negative numbers are used for spam documents, and the highest number for the most relevant pages. Alternatively, $2^{rel_k} - 1$ is used as gain function to reward the most relevant documents even more. With the standard discount, the formula for DCG is:

$$DCG = \sum_{k=1}^{K} (2^{rel_k} - 1) \times \frac{1}{\log_2(k)} \tag{2.2}$$

To allow aggregation over multiple topics in a test collection, DCG scores are usually normalized by an idealized DCG score, computed for the ideal ranking (i.e., sorted from most relevant to least relevant). This is referred to as normalized DCG or nDCG in short.

2.1.4 Further Reading

IR is a very broad research field and it is beyond the scope of this thesis to discuss all facets of it in detail. In the following sections we will discuss facets of IR that are relevant for each of the research themes. This is by no means intended as a full overview of IR research. Instead, we refer to four excellent textbooks on IR [12, 40, 54, 134]. Of these, the textbook by Baeza-Yates and Ribeiro-Neto [12] is the oldest and covers modern IR topics such as web search, user interface design and multimedia. Manning et al. [134] place particular emphasis on fundamental IR models, whereas the books by both Croft et al. [54] and Büttcher et al. [40] emphasize the practicalities of building a search engine and an inverted index in particular, covering topics such as compression, dynamic indexing and efficiency.

2.2 Studying News Collections

In the first research theme, *studying news collections*, we look at researchers who study large collections of news articles. Inspired by information seeking tasks from the social sciences and humanities, we propose new IR algorithms for news search. The news domain is traditionally a very active domain for IR research, where many new IR tasks and ideas were pioneered. Some examples include detecting and tracking topics [5], exploring either themes in news over time [93] or how the past is remembered [10] and searching without an explicit query [96]. Particularly challenging in news search is to understand the context of articles, as news is versatile, fluid, and background information is often important but limited. News stories are volatile and often presented without much context or with implicit context in case of a developing news story, where it is typically assumed that news consumers are aware of current affairs. This calls for IR methods that are flexible and for new approaches to modeling context and finding connections between stories.

Over the past decade, IR research in the news domain has witnessed a rapid broadening. From a strong focus on analyzing facts the field is broadening to also include more subjective aspects, such as opinions and sentiment [171], human values [70], argumentation [170] and user experiences from online forums [107]. This opens up new avenues for research, not just within computer science. Emerging fields such as digital humanities [195] and computational social science [123] aim to marry digital and computational methods with research methodologies from the humanities.

In this section, we discuss the background for our work within this theme, building on the IR concepts introduced in Section 2.1. In §2.2.1, we discuss how researchers study large news collections, with a particular emphasis on the methodology employed by historians. We connect this with two concepts from IR: exploratory search in §2.2.2 and automatically creating connections between collections in §2.2.3.

2.2.1 Studying News Archives in the Humanities

A large amount of digital material has become available to study our recent history, ranging from digitized newspapers and books to, more recently, web pages about people, places and events. Each collection has a different perspective on what happened and this perspective depends partly on the medium, time and location of publication. The availability of vast amounts of publicly accessible digital data motivates the development of techniques for large-scale data analysis and brings about methodological changes in disciplines that are shifting from data-poor to data-intensive [123, 195].

Understanding word meaning by studying how words are being used and how their usage changes has a tradition that goes back at least half a century. The field of statistical semantics focuses on the statistical patterns of words and how they are used [224]. The underlying assumption is that “a word is characterized by the company it keeps” [68]. Recent innovations have allowed statistical semantics methods to be applied to larger and larger datasets. Michel et al. [147] have used these methods on millions of digitized books (∼4% of all books ever published), to observe cultural trends. Au Yeung and Jatowt [10] studied how the past is remembered on a large scale. They find references to countries and years in over 2.4 million news articles written in English and study how these are referred to. Using topic modeling they find significant years and topics and compute similarities between countries. More recently, Kenter et al. [113] looked at vocabulary shifts over time and proposed methods to track a topic with changing vocabulary using a small set of seed words. In the remainder of this subsection, we look in detail at studying news within the research field that is of relevance to the first research theme: the humanities.

Historical research methodology. So far, the humanities have profited only marginally from large-scale digital collections that possibly span decades or even centuries. As the American Council of Learned Societies speculated, this may be a result of the distinct nature of data in the humanities, which are often historically specific, geographically diverse, and culturally ambiguous [52]. Hence, a qualitative analysis of documents is a first choice. In historical research one often closely studies a manually determined sample of the material [51, 215, 230]. These traditional historical research methods can be used for studying large-scale digital collections, but they impose substantial limitations. Adapting these research methods and combining them with computationally-based methods—for both document selection and the analysis of the selected documents—may yield new research questions for historians.

Looking at the information seeking behavior of historians when collecting relevant material, Duff and Johnson [59] identified four types of tasks that they employ: (1) orientation, (2) known material search, (3) building contextual knowledge and (4) identification of relevant material. Orienting to new collections happens throughout the research process and has the aim of getting a sense of what a collection has to offer. Known material search occurs when a researcher has a specific document in mind and seeks to locate this. Building contextual knowledge to ground further research is at the core of the historical research methodology. Historians critically regard sources in their context, considering aspects such as date, authorship and localization in order to assess the credibility of a source [78]. When multiple sources are considered, each might provide an interestingly different perspectives on a historical events, expressing views at the time of writing or even a subjective view of the author. For example, Lensen [127] analyzes two contemporary novels about the second World War and shows that the writers have a different attitude towards the war than previous generations when it comes to issues of perpetratorship, assignation of blame and guilt. Lastly, contextual knowledge such as the author and creation date is extensively used in the last information seeking task of identifying relevant material.

Document selection. Corpora available for historical research are often too large to be examined entirely. Researchers select subsets and closely examine only the selection. This leads to the fundamental choice: what to select? The most common approach to do this in historical research is through sampling. We describe three examples.

Van Vree [215] studied Dutch public opinion regarding Germany in the period 1930–1939. This was one of the first studies that explored the possibility to use media history as a form of mentality history. Van Vree [215] assumed newspapers to be the most important mass media at that time and selected four newspapers that represented major population groups (such as Catholics and Protestants). All issues of these newspapers were browsed manually, yielding a selection of almost 4000 articles that expressed an opinion on Germany. Neutral press, with a marketshare of about 45%, were not considered. Witte [230] followed a similar approach to study the image of Belgium during the Belgian Revolution. Six newspapers from different cities and political signatures were manually selected and browsed, keeping only 350 articles that expressed an opinion, possibly omitting many more. Condit [51] studied public expositions on heredity and genetics; based on indexes provided by publishers, 650 articles were selected from a period of 95 years. Considering the dynamic and often inaccurate nature of indexes, one can question the representativeness of this sample. Moreover, only public expositions were studied, implicit assumptions regarding heredity were not studied.

These studies provide important insights on public opinion, but there are important practical and theoretical disadvantages to the methodological approach they employ. One

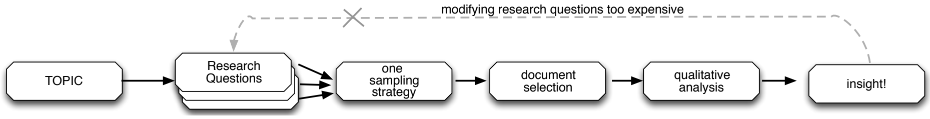


Figure 2.2: A graphical representation of the document selection process using manual sampling.

can argue that not all relevant articles were selected, yet checking this would mean redoing the entire laborious selection process. Insight gained from inspecting the collection cannot be used to obtain a more representative sample. Even though methods to explore and study large collections have come a very long way, this arguably subjective and rigid method of manual sampling is common practice.

We have modeled manual sampling in Figure 2.2. Given a research topic, a historian poses research questions that are then used to form a single sampling strategy. This strategy determines which documents to include in the selection. The researcher then develops an insight into the topic based on a qualitative analysis of the selection. This is a one way process: exploring new research questions means redoing the entire laborious process of sampling. In Chapter 3 we return to this process and propose an alternative approach based on exploratory search techniques.

2.2.2 Exploratory Search

Exploratory search is a form of information retrieval where users start without a clear information need. They do not know precisely what they want before they start their search, nor where they can find it [135]. Instead, they prefer to explore a collection to uncover new associations between documents. Here, users explore the collection, and iteratively fine-tune their queries until they find what they are looking for. Exploratory search systems therefore try to interactively and iteratively guide the user to interesting parts of the collection. Exploratory search interfaces often try to provide a quick overview, while allowing users to quickly zoom into details. Providing this quick overview can be done by visualizing the information that was retrieved [34, 35, 159]. When exploring a large collection, this can help maintain a sense of context [234].

When dealing with a temporally organized collection in exploratory search, there is a clear emphasis in interest on the evolution and development of documents, topics and word usage. In collections spanning long periods, time can be an important retrieval cue and insight into word usage is an important aspect to understanding the collection that is explored. Shneiderman [200] identified temporal data as a basic data types most relevant for information visualization. Visualizing time-oriented data has been extensively studied; for an overview see [3].

An interactive exploratory search system can help researchers such as historians to set up systematic search trails: the tooling helps them interpret and contrast the document sets returned. In Chapter 3, we describe in more detail how such a system can help a researcher explore collections, specifically when the documents in these collections are automatically connected across and within collections.

2.2.3 Connecting Collections

With more collections becoming digitally available, researchers have increasingly attempted to find connections between collections. It is common to link items based on their metadata. When items are annotated with concepts from a thesaurus or ontology, ontology alignment [158] can be used to infer links between items. An example is MultimediaN E-culture, where artworks from museums were connected based on alignments between thesauri used to annotate the collections [194]. An approach that does not rely on the presence of metadata is to infer links between collections based on textual overlap of items. For example, Bron et al. [33] study how to create connections between a newspaper archive and a video archive. Using document enrichment and term selection they link documents that cover the same or related events. Similarly, in the PoliMedia project [116] links between political debates and newspapers articles are inferred based on topical overlap. Both used publication date to filter documents in the linking process. However, how to score matches using these dates and combine them with temporal references is an open problem. Alonso et al. [7] survey trends in temporal information retrieval and identify open challenges, which include how to measure temporal similarity and how to combine scores for textual and temporal queries. We address these two challenges in Chapter 3.

To support media studies researchers, Bron et al. [35] propose a subjunctive search interface that shows two search queries side-by-side. They study how this fits into the research cycle of media studies researchers. They find that when using the proposed interface, the researcher explore more diverse topics and formulate more specific research questions. ManyPedia [137] allows users to explore different points of view by showing two Wikipedia articles from different languages side-by-side. A similar approach was used to synchronize cross-lingual content [152] on Wikipedia.

In Chapter 3, similar to the work described above, we provide exploratory search tools that emphasizes different perspectives. Our work differs in that we provide an end-to-end solution, with an emphasis on connecting multiple collections to explore and compare them

2.2.4 Studying News in the Social Sciences

In Chapter 4, we turn our attention to the social sciences. In this field, there is a growing trend of applying computational thinking and computational linguistic approaches. In particular, IR technology is proving to be a useful but underutilized approach that may be able to make significant contributions to research in a wide range of social science topics [47]. One particular topic in which this is happening is the study of news and its impact. Early examples focus mostly on analyzing factual aspects in news, e.g., Meijer and Kleinnijenhuis [144] analyzed the impact of news on corporate reputation by measuring the amount of news about specific issues. Increasingly, however, we are also seeing the use of IR technology to analyze more subjective aspects of news for the purposes of social science research [123].

Thematic content analysis. In the social sciences, mass communication (e.g., news) is often studied through a methodology called content analysis: “Who says what, to whom, why, to what extent and with what effect?” [121]. The aim of content analysis is to systematically quantify specified characteristics of messages. When these characteristics are complex, thematic content analysis can be applied: first, texts are annotated for indicator questions (e.g., “Does the item refer to winners and losers?”) and the answers to such questions are subsequently aggregated to support a more complex judgment about the text (e.g., an emphasis on a conflict). Content analysis is a laborious process, and there is a clear need for a computational approach. This approach can improve the consistency, efficiency, reliability and replicability of the analyses, as larger volumes of news can be studied in a reproducible manner, allowing the study of long-term trends.

Frames. In Chapter 4, we report on work aimed at analyzing the use of framing in news. Framing in news is the way in which journalists depict an issue in terms of a ‘central organizing idea’ [76]. Frames can be regarded as a perspective on an issue.

News coverage can be approached as an accumulation of “interpretative packages” in which journalists depict an issue in terms of a *frame*. Frames are the dependent variable when studying the process of how frames emerge (*frame building*) and the independent variable when studying effects of frames on predispositions of the public (*frame setting*) [193]. When studying the adoption of frames in the news, content analysis of news media is the most dominant research technique.

Using questions as indicators of news frames in manual content analysis is the most widely used approach to manually detecting frames in text. Indicator questions are added to a codebook and answered by human coders while reading the text unit to be analyzed [202]. Each question is designed such that it captures the semantics of a given frame. Generally, several questions are combined as indicators for the same frame. This way of making inferences from texts is also referred to as thematic content analysis [183].

Automatic or semi-automatic frame detection is rare. The approaches that do exist follow a dictionary-based or rule-based approach. For example, Ruigrok and Van Atteveldt [186] define search strings for the automatic extraction of a priori defined concepts in newspaper articles, and then apply a probabilistic measure to indicate associations between such concepts. Similarly, Shah et al. [198] first define “idea categories,” then specify words that reveal those categories, and finally, program rules that combine the idea categories in order to give a more complex meaning as a frame. The research in Chapter 4 describes an automatic approach for frame detection using text classification instead of a dictionary-based approach.

Four generic news frames. In Chapter 4, we focus on four commonly used generic news frames, originally proposed by Semetko and Valkenburg [197]: the conflict frame, human interest frame, economic consequence frame and morality frame.

The *conflict frame* highlights conflict between individuals, groups or institutions. Prior research has shown that the depiction of conflict is common in political news coverage [154], and that it has inherent news value [74, 218].

By emphasizing individual examples in the illustration of issues, the *human interest frame* adds a human face to news coverage. According to Iyengar [102], news coverage can be framed in a thematic manner, taking a macro perspective, or in an episodic manner, focusing on the role of the individual concerned by an issue. Such use of exemplars in news coverage has been observed by several scholars [154, 197, 235] and connects to research on personalization of political news [102].

The *economic consequence frame* approaches an event in terms of its economic impact on individuals, groups, countries or institutions. Covering an event with respect to its consequences is argued to possess high news value and to increase the pertinence of the event among the audience [75].

The *morality frame* puts moral prescriptions or moral tenets central when discussing an issue or event. Morality as a news frame has been studied in various academic publications and is found to be applied in the context of various issues such as, for example, gay rights [157] and biotechnology [32].

In Chapter 4, we present and evaluate an ensemble-based classification approach for frame detection in news. To the best of our knowledge, this is the first work in which statistical classification methods are applied to this central issue in studying media. Furthermore, we investigate whether explicitly modeling the thematic content analysis approach described above improves performance.

2.3 Struggling and Success in Web Search

In the second research theme of this thesis, we turn our attention to the domain of web search and the behavior of users that cannot find what they are looking for. When searchers experience difficulty in finding information, their struggle may be apparent in search behaviors such as issuing numerous search queries or visiting many results within a search session [11]. Such long sessions are prevalent and time-consuming (e.g., around half of Web search sessions contain multiple queries [201]). Long sessions occur when searchers are exploring or learning a new area, or when they are struggling to find relevant information [135, 228]. Methods have recently been developed to distinguish between struggling and exploring in long sessions using only behavioral signals [92]. This is important since struggling is prevalent in long tasks, e.g., Hassan et al. [92] found that in 60% of long sessions, searchers' actions suggested that they were struggling.

Related research has targeted key aspects of the search process such as satisfaction, frustration, and search success, using a variety of experimental methods, including laboratory studies [11, 65], search log analysis [89], in-situ explicit feedback from searchers [72], and crowd-sourced games [2]. Such studies are valuable in understanding these important concepts, and yield insights that can directly improve search systems and their evaluation. Of particular interest to our research within this research theme (and therefore discussed below) is the extensive body of work on satisfaction and search success (§2.3.1), searcher frustration and difficulty (§2.3.2), and query reformulation and refinement (§2.3.3).

2.3.1 Satisfaction and Success

The concepts of satisfaction and success in search are related, but they are not equivalent. Success is a measure of goal completion and searchers can complete their goals even when they are struggling to meet them [89]. Satisfaction is a more general term that not only takes goal completion into consideration, but also effort and more subjective aspects of the search experience such as searcher's prior expectation [105]. Satisfaction has been studied extensively in a number of areas such as psychology [131] and commerce [168]. Within search, satisfaction and success can be framed in terms of search system evaluation, essential in developing better search technologies. In Section 2.1.3 we discussed IR approaches to system evaluation with relevance assessments. Kelly [112] comprehensively summarizes different methods for evaluating search systems with searchers.

At a session level, Huffman and Hochster [99] found a strong correlation between session satisfaction and the relevance of the first three results for the first query, the number of events and whether the information need was navigational. Hassan et al. [89] showed that it is possible to predict session success in a model that is independent of result relevance. Jiang et al. [105] found that it is necessary and possible to predict subtle changes in session satisfaction using graded search satisfaction. Most prior studies regard search tasks or sessions as the basic modeling unit, from which holistic measures (e.g., total dwell time [231]) can be computed. Beyond tasks and sessions, interest has also grown in modeling satisfaction associated with specific searcher actions [114, 222]. These estimates can then be applied to improve rankings for future searchers [91]. Insights into satisfaction and success are used to predict satisfaction for individual queries [72, 90] and for sessions [89, 99, 105].

2.3.2 Frustration and Difficulty

Related to satisfaction are other key aspects of the search process such as task difficulty and searcher frustration. These have been studied using a variety of experimental methods, including log analysis [92], laboratory studies [11, 65], and crowd-sourced games [2]. Feild et al. [65] found, in a user study in which participants were given difficult information seeking tasks, that half of all queries submitted resulted in some degree of self-reported frustration. Ageev et al. [2] provided crowd-workers with tasks of different levels of difficulty and found that more successful searchers issue more queries, view more pages, and browse deeper in the result pages. When searchers experience difficulty in finding information, their struggle may be apparent in search behaviors such as issuing numerous search queries, more diverse queries or visiting many results within a search session [11]. However, rather than being an indication of a user that is struggling, these longer sessions can be indicative of searchers exploring and learning [62, 228]. Hassan et al. [92] have recently developed methods to distinguish between struggling and exploring in long sessions using only behavioral signals. They found that searchers struggled in 60% of long sessions. Scaria et al. [192] examined differences between successful and abandoned navigational paths using data from a Wikipedia-based human computation game. They compared successful and abandoned navigation paths, to understand the types of behavior that suggest people will abandon their navigation task. They also constructed predictive models to determine whether people will complete their task successfully and whether the next click will be a back click (suggesting a lack of progress on the current path). The terminal click has also been used in other studies of search to better understand searchers' information goals [58] or point people to resources that may be useful to other searchers [227]. In this research theme, we focus on struggling sessions (that are thus likely to be unsatisfactory) to understand how some of them end up successful while others end up unsuccessful. We target traditional web search because of its prevalence. Recently, others have studied struggling and success in the context of engagement with intelligent assistants [105].

The studies presented in this section and those on searcher satisfaction are valuable in understanding the important concepts around search success, and yield insights and signals that can directly improve search systems and their evaluation [91]. They provide important clues on what searchers might do next, such as switching to a different search engine [85, 225] or turning to a community question answering service [129]. Ideally, a search engine would interpret these signals of struggling and frustration to provide personalized hints to help the searcher succeed. These hints can be learned from more successful and advanced users [2, 226] or provide examples that may work generically for some search intents, such as leading searchers to longer queries [1]. Moraveji et al. [153] showed that the presentation of optimal search tips, for tasks where they are known to have benefit, can have a lasting impact on searcher efficiency. Savenkov and Agichtein [191] showed that providing a searcher with task-specific hints improves both success and satisfaction. Conversely, generic hints decrease both success and satisfaction [153, 191], indicating that it is paramount to understand what a searcher is struggling with before providing hints.

2.3.3 Query Reformulation and Refinement

More detailed insight into searcher behavior can be found by analyzing query reformulations. Query reformulation is the act of modifying the previous query in a session (adding, removing, or replacing search terms) with the objective of obtaining a new set of results [90]. For this, a number of related taxonomies have been proposed [8, 84, 98, 122, 209]. Huang and Efthimiadis [98] surveyed query reformulation taxonomies and provided a mapping between these and their own approach. While they provide interesting insights into searchers trying to articulate their information needs, these approaches all focus on superficial lexical aspects of reformulation. Anick [8] examined usage and effectiveness of terminological feedback in the AltaVista search engine. No difference in session success was found between those using the feedback and those not using it, but those using it did continue to employ it effectively on an ongoing basis. A number of recent studies have shown that search tasks provide a rich context for performing log-based query suggestion [64, 111, 128], underscoring the importance of studying query reformulation in search tasks.

In Chapter 5, we employ large-scale log analysis and a crowd-sourced labeling methodology to provide new insights into the nature of struggling and what contributes to search success. Based on this, we propose a new taxonomy for intent-based query reformulation that goes beyond the surface-level lexical analysis commonly applied in the analysis of query transitions (see [98]).

2.4 Pro-active Search for Live TV

In the third and final research theme, we consider a pro-active search scenario, specifically in a live television setting, where we propose algorithms that leverage contextual information to retrieve diverse related content for a leaned-back TV viewer. Motivating this research theme is that the way people watch television is changing [79, 156, 180]. Increasingly, viewers consume broadcasts interactively and with additional content related to what they are watching. In a recent survey by Nielsen, around 70% of tablet and smartphone owners reported to use their devices while watching television [155]. Razorfish [180] found that 38% of "mobile multitaskers" access content that is related to the TV program they are watching.

In a live TV setting, the main challenge is to find content while a story is developing. For live TV, subtitles are typically available (provided for the hearing impaired). Using this textual stream of subtitles, we can automatically provide links to background information and generate search queries to find related content. Both tasks have unique demands that require approaches that need to (1) be high-precision oriented, (2) perform in real time, (3) work in a streaming setting, and (4) typically, with a very limited context. By leveraging the textual stream of subtitles, we cast these tasks as IR problems in a streaming setting. In the remainder of this section, we discuss related work on search in a streaming setting (§2.4.1), link generation (§2.4.2) and query modeling (§2.4.3).

2.4.1 Search in a Streaming Setting

Within this research theme, we leverage a textual stream of subtitles to find related content in real time. Early research on search in such a streaming setting setting has focussed on modeling the keywords in transcripts. For example, Brown et al. [36] analyze automatic speech recognition (ASR) transcripts in real time to produce keywords and topics. Similarly, Song et al. [204] study the task of keyword extraction from transcripts of meetings using a graph-based keyword extraction approach.

Henzinger et al. [96] propose an approach to find relevant news articles during broadcast news. Every 15 seconds, they produce a two term query, using a "history feature," consisting of the last three blocks of text. Recent work by Blanco et al. [25] on Yahoo! IntoNews builds and improves on the work of Henzinger et al. [96]. Their focus is on the problem of detecting a change of topic, for which they propose several segmentation approaches. After segmentation, queries are generated using a bag-of-word approach with TF.IDF scores for each term. In our research within this theme, we include the query modeling approach of Blanco et al. [25] as a baseline approach.

Generic methods to automatically segment text (e.g., TextTiling [95]) have been extensively studied. These methods have been shown to also perform well in the streaming settings of new event detection [6] and story segmentation, a subtask of topic detection and tracking (TDT). Specific segmentation approaches have been proposed for streaming settings similar to ours (such as [25, 96]). However, the subtitles that we work with in this research theme are generated from an auto-cue and thus contain markings indicating the start and finish of an item.

We propose two distinct approaches for searching in a streaming setting within this research theme. In Chapter 6, we propose a link generation approach for finding background information in real time. In Chapter 7, we model the terms in the stream of subtitles to generate queries that can be used to find related content. In the next sections, we discuss related work on both approaches, covering link generation and query modeling in Sections 2.4.2 and 2.4.3 respectively.

2.4.2 Link Generation

Automatically generating hypertext links has been studied for nearly two decades. Early work included defining a taxonomy of hyperlink types and applying string-matching methods for automatic links [4]. A typical use case is in linking news archives [82], more recently also across modalities [25, 33, 96]. This early work is closely related to the connecting collections approaches previously discussed in Section 2.2.3. The kind of link generation we consider in this research theme is commonly referred to as *entity linking*: phrases—consisting of a single term or sequence of terms—are automatically linked to entries in a knowledge base.

Entity linking. Entity linking facilitates advanced forms of searching and browsing in various domains and contexts. It can be used, for instance, to anchor the textual resources in background knowledge; authors or readers of a piece of text may find entity links to supply useful pointers [94, 141]. Another application can be found in search engines, where it is increasingly common to link queries to entities and present entity-specific overviews [13, 140]. More and more, users want to find the actual entities that satisfy their information need, rather than merely the documents that mention them; a process known as *entity retrieval*.

Early work on entity linking aimed to improve linking on Wikipedia, by finding missing links [69]. Fissaha Adafre and de Rijke clustered similar pages and identified candidate links to be added to a page. In entity linking, it is common to consider entities from a general-purpose knowledge base such as Wikipedia or Freebase, since they provide sufficient coverage for most tasks and applications. In this case, links are intended to be explanatory, by providing definitions or background information. Wikipedia is therefore a common target for entity linking. Automatic linking approaches using Wikipedia have met with considerable success [94, 140, 141, 148, 149]. Approaches for linking entities are not Wikipedia-specific, however. Recent developments in the Web of Data enable the use of domain or task-specific entities [119]. Alternatively, legacy or corporate knowledge bases can be used to provide entities [100]. Entity linking can also be employed as an concept or entity normalization step [37, 55, 148, 179].

Approach. Entity linking approaches generally consist of three steps. First, find all candidates for linking and then disambiguate possible targets. From this, select which candidates to link and pick the target to link to. These methods assume text is more or less clean and grammatically correct and rely heavily on context for link selection and disambiguation. This context has been modeled in different ways. Early work uses only *local* approaches to disambiguation [37, 148], looking at how well each link candidate fits in the text. This is done by comparing the content of the source document to the content the page to be linked to. In contrast, *global* approaches regard all link candidates for a document and then try to form a coherent set from these. These methods use relatedness measures based on the structure of Wikipedia [55, 149, 179]. Milne and Witten [149] balance a measure for how common a link is with how related it is to other links. Comparing all possible links for a document is computationally expensive, so a choice of what link candidates to consider is often made. Using only the few unambiguous link candidates such as in [149] dismisses many link candidates, while considering all makes the set subject to noise and impractically large [55]. For global approaches, semantic relatedness between articles is computed using a distance measure for the incoming and outgoing links and categories an article belongs to [55, 149, 179]. For streaming text, neither local nor global approaches for disambiguation are suited as linking is considered for just a chunk of text. These local and global approaches are computationally heavy, as there are many comparisons to be made for each link candidate.

Domains. Entity linking has been applied to many different specific domains. Jijkoun et al. [106] studied this as an entity normalization task in blogs and comments. He et al. [94] showed that applying entity linking to radiology reports did not yield satisfactory results and propose a sequential labeling approach, with syntactic features. More recently, work has gone into applying entity linking to short texts, such as queries [25, 67, 87, 139] and microblogs [141]. In these short texts, generic methods developed for documents fail, as grammar and context are virtually absent. Ferragina and Scaiella [67] developed a system called TAGME, designed specifically for short snippets of poorly composed text. For this they try to find collective agreement for the link targets using a voting scheme based on a relatedness score. The authors point out that computing this relatedness score is the most time-consuming step in their system. For entity linking of short texts, Meij et al. [141] found that, as a retrieval model for finding link candidates, lexical matching on anchor text performs better than lexical matching on title, language modeling and a document retrieval-based approach. In Chapter 6, we compare such an approach to other approaches on live TV subtitles [162] and show that this is indeed the best approach for finding link candidates.

Linking in multimedia content. The setting we consider in Chapter 6 has some particular challenges as it deals with (1) multimedia content and (2) a streaming setting. For the first challenge, two related tasks have been considered at the CLEF evaluation campaigns. VideoCLEF’09 [120] featured a task aimed at linking video content to related resources across languages. This task was framed as a known-item-task, where noisy ASR for Dutch was used to produce links to target English Wikipedia articles. The best performance was achieved by taking an off-the-shelf entity linking toolkit for documents [149]. The learning to rerank approach employed by the toolkit outperformed several retrieval approaches. The MediaEval Search and Hyperlinking task considers two related tasks: finding a known item in broadcast video and providing video hyperlinks for specific anchors. The first is a video retrieval task and for the second task participants focused on the multimedia aspect, e.g., by using image-based retrieval approaches.

For the second challenge of dealing with streaming textual content, we propose to model the textual context in a graph structure. Graph-based methods have been proposed for natural language processing (NLP) problems, such as word clustering [31], word dependency [210], text summarization [63] and topic modeling [138]. Erkan and Radev [63] show how a random walk on sentence-based graphs can help in text summarization. A well-known example of this idea of a random walk is PageRank [169]—one of the measures that we use (mentioned before in §2.1.2). PageRank measures the relative importance of a node in a graph. For webpages, this importance is computed by estimating the chance of ending up on a page, by random clicking on links. More generally, in a graph this importance is estimated by the chance of passing through a node when doing a random walk on a graph. In information retrieval, this approach has also been applied to graphs that consist of queries and documents [53] or only queries [27].

2.4.3 Query Modeling

A natural way of looking at search in streaming settings is to view it as a query modeling task, where the aim is to model an effective query based on a larger volume of text. We do so with the research presented in Chapter 7. The query modeling task has two distinct aspects: (1) query reduction and (2) content-based query suggestion. The former deals with reformulating long or descriptive queries to shorter and more effective queries. Driving this research is that shorter queries are not only more likely to retrieve more focused results, they are also more efficient to process [18]. The latter task is to generate queries and the methods used here are similar to those used for query reduction. We covered content-based query suggestion as applied to connecting digital collections (e.g., [33]) in Section 2.2.3. A broad range of other tasks for content-based query suggestions exists, for example, generating phrasal-concept queries in literature search with pseudo-relevant feedback [115]. It differs from content-based recommendation, where a user profile or user interactions are also available. Bendersky et al. [20] propose a hybrid approach using content-based heuristics and signals from user interactions to retrieve video suggestions for related YouTube videos. Such a hybrid approach is not feasible in our scenario within this research theme, as we have no user interactions that relate to the live TV content.

The state of the art in query modeling is formed by a family of related methods, e.g., [118, 125]. The typical approach is to produce query terms, score each, and generate one or more queries. Term scoring is often based on term features taken from large background corpora. Queries are selected based on query difficulty prediction features; these require an initial retrieval round, and as such are not feasible in real time. Lee and Croft [126] generate queries from textual passages by extracting noun phrases or named entities; a conditional random field is used to find selections that optimize retrieval effectiveness. In a web search setting, Kumaran and Carvalho [118] cast query reduction as a sub-query ranking problem. Given a long query, they generate all possible sub-queries for which they subsequently predict query quality and use learning to rank to select the best. Balasubramanian et al. [14] improve web search results by dropping query terms and estimating reductions in query difficulty. They generate reduced queries and predict query difficulty. Based on the difference in difficulty between reduced and original queries, they select the best reduced query. The original query is either replaced or the results for the original query and the best reduced query are combined.

Methods to improve the retrieval effectiveness of descriptive queries are proposed in [19, 124, 125]. Lease et al. [124, 125] generate queries of up to six terms by sampling term weights and then computing a metric to learn query term weights. Bendersky and Croft [18] extract key concepts from verbose queries and demonstrate that integrating these into queries significantly improves retrieval effectiveness on newswire and web collections. They follow up on this work in [19] with a parameterized query expansion approach, that is particularly effective for verbose queries. Similar features are used in [19, 125, 126]; the first two obtain comparable retrieval scores on descriptive queries. For further reference, we refer to Gupta and Bendersky [86], who surveyed these and other approaches for improving the retrieval effectiveness of so-called verbose queries using query modeling. Our work in Chapter 7 builds upon these query modeling approaches. The work differs from these stationary query modeling approaches in that we explicitly model the dynamic nature of streaming sources and generate more complex queries (e.g., using different weights for different fields).

Part I

Studying News Collections

3
Exploration of Historical Perspectives across Collections

Part I of the thesis focuses on the first of three research themes: *studying news collections*. This chapter is the first of five research chapters in the thesis and we start our investigation by focusing on how researchers study and explore large collections. A huge amount of digital material has become available to study our recent history, ranging from digitized newspapers and books to, more recently, encyclopedic web pages about people, places and events [195]. With that, the wish for researchers grows to explore and study these collections and in particular to study the digitized news articles in their context. Above, in Section 2.2.1, we detailed how humanities scholars study news archives. In this field, despite the vast amounts of publicly accessible digital data, qualitative analysis of documents is still a first choice. This may be a result of the distinct nature of data in the humanities, which are often historically specific, geographically diverse, and culturally ambiguous [52]. In historical research, one closely studies an often manually determined sample of the material [51, 215, 230]. These traditional historical research methods can be used for studying large-scale digital collections, but they impose substantial limitations.

As a motivational use case to the research in this chapter, we first describe the intertwined development of a method to select documents and an exploratory search system to access a digital archive spanning several centuries, designed to provide a historian with valuable insight. The system supports historians to set up systematic search trails and helps them interpret and contrast the result sets returned. By exploring word associations for a result set, inspecting the temporal distribution of documents, and by comparing selections historians can make a more principled document selection. Looking at the four information seeking tasks that historians employ according to Duff and Johnson [59], this document selection process is a combination of three of the four: (1) orientation, (2) known material search and (3) relevant material identification (described in Section 2.2.1).

After this motivational use case, we study in more detail the fourth information seeking task: (4) building contextual knowledge to ground further research [59]. Humanities scholars critically regard historical sources in their context considering aspects such as date, authorship and localization in order to assess the credibility of a source [78]. When multiple sources are considered, each might provide an interestingly different perspective on a historical event, expressing views at the time of writing or even a subjective view of the author. Connections between collections help historians contextualize the sources.

In this chapter, we propose a method to automatically create connections between heterogeneous digital collections through temporal references found in documents as well as their textual content. Digitized collections pose interesting challenges and will require additional preprocessing. Where news articles have a clear temporal footprint, other sources such as encyclopedic articles might not. For these, temporal references can be extracted and leveraged in combination with the textual content to find related articles. We cast this as an information retrieval (IR) task and ask:

RQ1 Can we effectively extract temporal references from digitized and digital collection and does leveraging these temporal references improve the effectiveness of retrieving related articles?

To illustrate how our algorithms could be used to automatically create connections between digital and digitized collections, we introduce a novel search interface to explore and analyze the connected collections that highlights different perspectives and requires little domain knowledge. The remainder of this chapter is organized as follows. First, we revisit the document selection process as a motivational use case in Section 3.1. Next, we describe our approach to connecting digital collections in Section 3.2. Section 3.3 describes our exploratory and comparative interfaces. We provide a worked example in Section 3.4, after which we conclude in Section 3.5.

3.1 Revisiting the Document Selection Process

When studying large collections, historians select subsets and closely examine only their selections. This leads to the fundamental choice: what to select? The three examples [51, 215, 230], detailed above in Section 2.2.1, employ the most common approach: *sampling*. These three studies provide important insights on public opinion, but there are important practical and theoretical disadvantages to the methodological approach they employ. One can argue that not all relevant articles were selected, yet checking this would mean redoing the entire laborious selection process. Insight gained from inspecting the collection cannot be used to obtain a more representative sample. This arguably subjective and rigid method of manual sampling is common practice. The selection of documents from a collection spanning over four centuries calls for an alternative. We have modeled manual sampling in the top part of Figure 3.1. Given a research topic, a historian poses research questions that are then used to form a single sampling strategy that determines which documents to include in the selection. The researcher then develops an insight into the topic based on a qualitative analysis of the selection. This is a one way process: exploring new research questions means redoing the entire laborious process of sampling.

Without dismissing the traditional document sampling methods, we propose an alternative document selection strategy. In *semantic document selection* (see the bottom part of Figure 3.1) research questions are associated with queries against a collection. This takes away the limitation of having a single sampling strategy. A query can consist of keywords, a specific time period, a particular document source, or any combination. Each query yields a document selection, with no laborious sampling needed. Through visualizations, new insights can be gained from an initial selection. This can lead to an

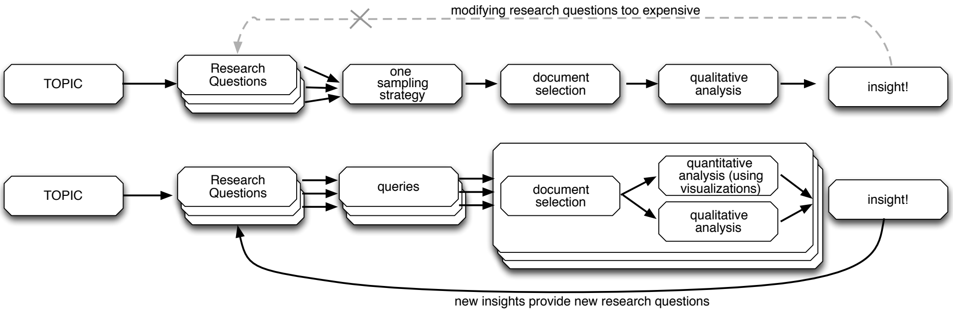


Figure 3.1: Two implementations of the document selection process, contrasting the manual sampling method (top) with semantic document selection (bottom).

improved query and, therefore, a more representative document selection. This can be done by exploring word associations and metadata and through a visualization of the number of documents over time. A clear benefit is that the historian can use the gained insights to investigate new research questions. Moreover, comparing document selections using quantitative analysis helps to validate these selections, making them less biased and more representative. With manual sampling, validating the document selection is impractical or even impossible as replicating the manual selection process is too time intensive.

Semantic document selection as a methodology for historical research on large repositories addresses three problems of traditional manual sampling: representativeness, reproducibility and rigidness. Word associations improve representativeness of the document selection as these associations are produced from the data, not from prior knowledge. Comparing selections and inspection of specific timespans in the data further supports the researcher's understanding of the representativeness of a document selection. It allows document selections to be reproducible and removes the rigidness that stems from a single sampling strategy. Associations, longitudinal search and comparisons allow the researcher to return to document selection with new insights, at any time.

ShoShin. To support the document selection process, we developed ShoShin, an exploratory search interface that guides the user to interesting bits of information, leveraging the fact that the users are experts on the topic of interest. For details on the architecture of ShoShin, we refer the reader to the original publication [159]. We consider the digital newspaper archive of the Koninklijke Bibliotheek (KB), that consists of about one hundred million newspaper articles and is described in detail later in this chapter. Figure 3.2 shows an abstract overview of the interface. ShoShin provides the user not just with a list of relevant documents, but also with visualizations that allow inspection of, and navigation through, the document selection.

The visualization of word associations allows historians to glance over the content of the document selection. This visualization is a term cloud based on the relative frequencies of the words occurring in documents within a selection. Clicking on words in the cloud modifies the selection. A temporal distribution visualization allows historians to discover patterns in publication dates. This visualization is a histogram of publication dates that can be explored interactively. To enable quick recognition of atypical patterns,

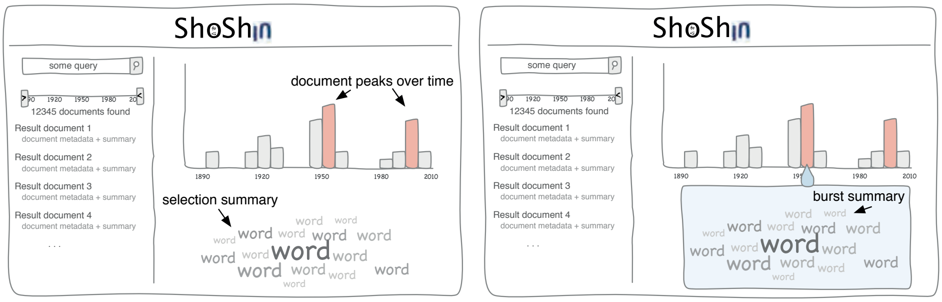


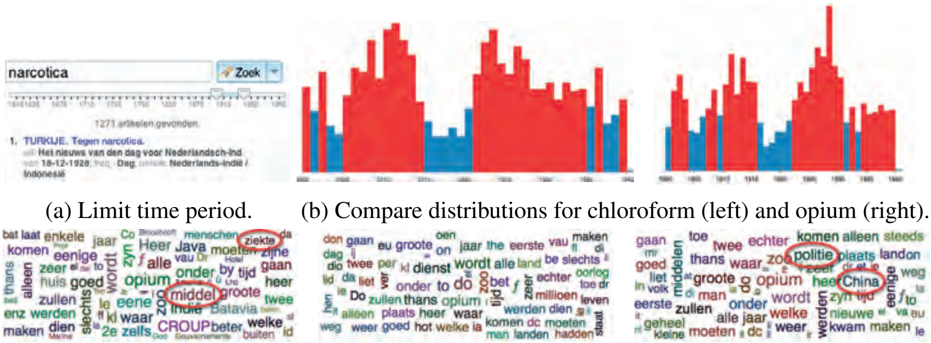
Figure 3.2: Interface sketches of ShoShin. A user enters a query (top left), resulting in a list of relevant documents, a term cloud and temporal distribution visualization (left). Users can click on bursts to see word associations of that burst (right).

bursts within the histogram—time periods where significantly more documents were published compared to neighboring periods—are highlighted. Clicking on a burst yields a visualization of word associations of that burst alone and a list of documents contained within that burst. This allows the historian to get an in-depth understanding of what each burst is about. Together, these interactions facilitate exploration of the document selection in order to detect patterns, improving the representativeness of the selection.

A worked example. To illustrate how a historian would use Shoshin, we report on one of several case studies with individual historians. Our subject, a senior historian, wants to analyze the public opinion on drugs, drug trafficking, and drug users, as represented in newspapers, in the early twentieth century (1900–1940). He wants to know whether the view on drugs is predominantly based on medical aspects (addictions, health benefits) or on social aspects (crime). How does our subject use ShoShin? First, he needs to create a lexicon of terms related to drugs. The term *narcotica* is a Dutch umbrella term for several narcotics. The actual terms describing narcotics may have changed over time and not all may be known to the historian. For a high recall of documents, a lexicon is required that captures all possible relevant terms. When a researcher uses his *domain knowledge* to create a list of words, ShoShin supports the researcher to find terms that are not readily available to him by showing a *term cloud* based on all retrieved documents (see Figure 3.3c for term clouds for drug related queries). The historian can expand the original query with terms he recognizes as drugs.

To inspect the representativeness of the document selection, the historian looks at the temporal distribution of documents. He sets the time period to 1900–1940 (see Figure 3.3a), queries for several names of drugs and compares the resulting temporal distributions. Based on his domain knowledge he marks key events, like the Opium Treaty from Shanghai (1912), the introduction of Dutch Opium laws (1920) and the tightening thereof (1928). From this known material search task [59], he then concludes that before the Dutch Opium laws came into effect the term *chloroform* was dominantly used; afterwards, the terms *opium*, *heroïne*, and *cocaïne* are more prominent (see Figure 3.3b).

To gain a better understanding of the aspects associated with drugs, the historian looks at what terms were associated with the drugs over time, by examining the associated term cloud. In this orientation task [59], he compares term clouds of several time periods at


narcotica
Zoek
1271 artikelen gevonden.
1. TURKIJE. Tegen narcotica.
Het nieuws van den dag voor Nederlandsch-Indië
18-12-1928
Dag
Nederlands-Indië / Indonesië
ziekte
middel
politie
China
opium

(a) Limit time period. (b) Compare distributions for chloroform (left) and opium (right).

(c) Explore associations for the years 1902, 1920 and 1927 (from left to right).

Figure 3.3: Screenshots from a case study of a historian using ShoShin.

several scales in time. These associated term clouds (Figure 3.3c) show a shift from health issues (*geneesmiddelen*, *vergiften*, *wetenschap*, *apotheken*[1]) to crime related issues (*politie*, *smokkelhandel*, *gearresteerd*[2]). By inspecting the actual word counts, the historian can find quantitative evidence for an increased use of the terms associated with *narcotica* after the Dutch Opium laws came into effect and that they are decreasingly associated with health related terms and increasingly associated with crime related terms. He can then continue with the task of relevant material identification [59] to support this observation and study it in more detail.

This example suggests that semantic document selection, where a selection is associated with a meaningful query, fits well in historical research methodology as an alternative to manual sampling, improving the representativeness and reproducibility of document selection and, thereby, the validity of the conclusions drawn.

3.2 Connecting Digital Collections

The motivational use case presented above describes how researchers from the humanities select subsets of large collections for close examining. When closely examining a document selection, historians critically regard these documents in their context. They consider aspects such as date, authorship and localization in order to assess the credibility of a source [78]. This can be further supported by considering multiple sources. In the remainder of this chapter, we propose an approach to support historians in the task of building contextual knowledge [59], by automatically creating connections between documents across collections. In this section, we present an algorithmic approach to connect multiple heterogeneous collections. After describing and evaluating a solution for finding related articles, we describe how these algorithms can be used in an interactive search application that supports researchers (such as historians) in studying different perspectives in the connected collections.

[1] English: medications, poison, science, pharmacies

[2] English: police, smuggling, arrest

In the remainder of this chapter, we focus on the second World War (WWII), as this is a well-studied and defining event in our recent history. We provide tools for selecting, linking and visualizing WWII-related material from collections of the NIOD,[3] the National Library of the Netherlands, and Wikipedia. These different collections tell different stories of events in WWII and there is a wealth of knowledge to be gained by comparing these. Reading a news article on the liberation of the south of the Netherlands in a newspaper collaborating with the occupiers gives the impression that it is just a minor setback for the occupiers. A very different perspective on the same events emerges from articles in an illegal newspaper of the resistance, leaving the impression that war is ending soon. To complete the picture, these contemporary perspectives can be compared to the view of a historian who—decades later—wrote fourteen books on WWII in the Netherlands and to the voice of thousands of Wikipedians, all empowered with the benefit of hindsight. Although we focus on events and collections related to WWII, our approach and application can be applied to other collections and topics.

In the remainder of this section, we describe in detail how we connect multiple heterogeneous digital collections. Simply put, we connect documents from different collections via implicit events using time and content. If we consider two newspaper articles from different newspapers, but published on the same day and with considerable overlap in content, we can infer that it is likely that they cover the same event. Newspaper articles are associated with a clear point in time, the date they were published. However, not all collections have such a clear temporal association. We therefore infer these associations from temporal references (i.e., references in the text to a specific date). We describe how we extract the relevant dates in Section 3.2.2 and evaluate this in Section 3.2.3. Our novel approach to connecting collections can deal with these extracted temporal references. We present and validate this approach in Section 3.2.4 and Section 3.2.5 respectively. First, we describe our collections.

3.2.1 Collections

We connect three heterogeneous collections, each representing a different kind of data source: (1) a digitized collection of around 100 million Dutch newspapers articles, spanning four centuries, (2) the encyclopedic articles of the Dutch Wikipedia, (3) the digitized book series *Het Koninkrijk der Nederlanden in de Tweede Wereldoorlog*[4] by historian Loe de Jong. Table 3.1 provides an overview of the size and characteristics of each collection. For conciseness, we will refer to these three collections as the *Newspapers*, *Wikipedia* and *Loe de Jong* collections respectively in the remainder of this chapter.

The newspaper archive of the National Library of the Netherlands consists of around 100 million digitized newspaper articles, processed using optical character recognition.[5] Each newspaper article has consistent metadata, including publication date. To focus on the relevant articles we filter out articles published before 1933 or after 1949. The books of Loe de Jong are a standard reference work about WWII in the Netherlands, consisting of 14 volumes and published between 1969–1988 in 29 parts, all recently digitized [56]. Early parts focus on chronicling events in specific years whereas some later ones focus on

[3] NIOD Institute for War-, Holocaust and Genocide Studies, `http://www.niod.nl`.

[4] In English: *The Kingdom of the Netherlands during WWII.*

[5] The collection is available for browsing via Delpher, `http://delpher.nl/kranten`.

Table 3.1: Statistics for the three collections.

	Newspapers	Wikipedia	Loe de Jong
Number of documents	21,456,471	2,699,044	1,600
Average number of terms per document	105	91	1,776

specific themes. Each section was treated as a new document. Note that the documents in this collection are substantially longer than those in the two other collections.

3.2.2 Extracting Temporal References

We connect the three heterogeneous collections presented above via content and time. For the digitized newspaper collection, we use the publication date of an article. This metadata is clean and checked by the National Library. However, for Wikipedia articles and the books of Loe de Jong no clear dates are present. We extract the dates that these articles refer to through a process of temporal tagging, i.e., we extract references to dates from the article content. For this, we use a custom pipeline for xTAS,[6] an extendable toolkit for large-scale text analysis. We release this custom pipeline as open-source software[7] and contributed the temporal tagger to the xTAS project. Concretely, our approach for extracting dates consists of three steps: (1) *preprocessing*, (2) *temporal tagging* and (3) *aggregating*.

In the preprocessing step, we normalize the text and prevent common errors we encountered in the subsequent temporal tagging. For Wikipedia articles, we remove all special syntax, used for formatting and creating tables. We extract the textual content of a Wikipedia article using a MediaWiki syntax parser.[8] For the Loe de Jong collection, we remove XML tags and keep only the textual content. This textual content has been obtained from book scans using optical character recognition (OCR). Therefore, it can contain errors in the recognition of terms. We process this digitized text to remedy common OCR errors that we encountered, in particular errors that influence temporal tagging. For example, the numbers 0, 1 and 5 are commonly confused for °, I and S.

For both collections, we also use simple textual replacement rules to prevent common errors we found after an initial evaluation of temporal tagging on our data. A common short-hand way of referring to years is to use an apostrophe followed by only the last to digits: the period ’40–’45. As this gives no information on the century being referred to, such a reference is typically ignored by a temporal tagger. However, these references often refer to the 1900s and given that the topic of most of our document collection (WWII), we resolve a reference as above to the period 1940–1945.

After preprocessing, we analyze the content of each Wikipedia article and each document in the Loe de Jong collection using the multilingual cross-domain temporal tagger Heideltime [206]. The aim of a temporal tagger is to find all mentions of time and dates and to pinpoint these as exact as possible to a specific point in time. The output of

[6] eXtensible Text Analysis Suite, available on `http://xtas.net`.

[7] The source code is available on `https://bitbucket.org/qhp`.

[8] Mwlib extracts raw text from wiki articles, see `http://github.com/pediapress/mwlib`.

Table 3.2: Precision and recall of the extracted temporal references as measured by comparing the automatically annotated date references to those annotated by judges.

Collection	Annotated Documents	Annotated References	Unique References	Precision	Recall
Wikipedia	50	834	609	98.27%	86.72%
Loe de Jong	20	713	469	63.86%	68.04%
Loe de Jong	*without preprocessing for a.o. OCR errors*			36.86%	30.48%

Heideltime is a set of temporal references normalized according to the TIMEX3 annotation standard [176]. Temporal tagging of historical documents is particularly challenging due to the fact that temporal expressions are often ambiguous and under-specified. For example, *"in the 1930s"* refers to the time interval 1930–1939, while *"august 1945"* refers to the entire month of August, 1945. For most of these challenges, Heideltime is able to extract and normalize temporal references, even if they are under-specified.

The final step to extracting dates is aggregating all temporal references to the document level. We separate each temporal reference based on the annotation granularity (i.e., exact day, specific month or only year). We store and treat them differently both in connecting collections (see Section 3.2.4) and in the exploratory visualizations (see Section 3.3.1).

3.2.3 Evaluating Temporal Reference Extraction

We validate our approach to extracting date references with an experiment. We take 50 random documents from the Wikipedia collection and 20 from the Loe de Jong collection. Five judges annotate all date references within the documents. Table 3.2 details statistics on the annotated documents. Despite the substantially longer documents in the Loe de Jong collection (see Table 3.1), the narrative structure of the books leads to less frequent temporal references than the encyclopedic style of Wikipedia. We compute inter-annotator agreement over five doubly annotated Wikipedia documents and three documents from the Loe de Jong collection. We observe 97% agreement on the set of unique dates referenced, signaling excellent agreement among the human annotators. We measure the accuracy of the extracted temporal references by comparing the automatically annotated date references to those annotated by the judges. The results in terms of precision and recall are presented in Table 3.2.

On the Wikipedia collection, we observe a mean precision of 98% on the set of all automatically extracted dates. For unique annotated dates, which is what we use for temporal similarity, we observe a recall of 87%. These scores are comparable to what is reported on standard datasets [206] and signals that the task of extracting dates from Wikipedia articles is well suited to be done automatically.

On the Loe de Jong collection, we obtain substantially lower precision of 64% and recall of 68%. The sections of these books pose two distinct challenges for temporal tagging. First, as the books are digitized using OCR, there are errors in detected terms, including in parts of dates. Our preprocessing approach to remedy some of the common errors has doubled both precision and recall (up from 37% and 30% respectively). The second challenge is more difficult. The books of Loe de Jong are written in a narrative style, where temporal references are often (partially) implicit. For example, a section on famine in the winter of 1944–1945 (referred to as the “hunger winter”) often only refers only to days in these winter months, without referring to a year. Given the topic, a reader knows that January 15th refers to January 15th, 1945, but for an automatic approach, this is rather difficult to infer. In fact, half of the fourteen books indicate in the title that they cover only a specific period. Simply adding the book title to each of the sections would not resolve this, as the title might be a vague reference and temporal references later in the sections will overrule this.

Improving the accuracy of temporal reference extraction on such a collection poses interesting future work for information extraction researchers. Given the length of the documents and thus large number of temporal references, the level of accuracy we obtain after preprocessing is sufficient for ours and similar applications. The extracted temporal references for the Loe de Jong collection are published as Linked Open Data.[9] This enrichment allows for new types of temporal analysis of the collection.

This approach for temporal reference extraction is the first step in connecting these three heterogeneous collections. Using the temporal references for Wikipedia articles and sections of the books of Loe de Jong, combined with the publication dates of newspaper articles, we can find subsets of documents for a specific time period that were either published within that period or refer to a point in time within that period. However, this does not yet mean that all the documents in the subsets are topically related. For this, we also need to look at the content of the document.

3.2.4 Combining Temporal and Textual Similarity

In Section 2.2.3, we described existing approaches for creating connections between collections. A common approach that does not rely on the presence of metadata is to infer links between collections based on textual overlap of items [33, 116]. When temporal metadata is available, this is typically used to either filter documents [33, 116] or by using a *document prior* [173]. A temporal document prior gives preference to recent documents and is computed irrespective of the document content. The retrieval score of a document is then computed by multiplying the textual retrieval score with the temporal document prior. We build on the concept of a temporal document prior and extend it to leverage automatically extracted temporal references.

To estimate whether two documents refer to the same implicit event, we combine textual similarity with temporal similarity. We measure textual similarity based on a common approach for connecting collections [33, 116] and for content-based query suggestions [25, 96, 115] (see also Section 2.2.3 and Section 2.4.3 respectively). In this approach, textual similarity is computed as the Manhattan distance over document terms in a TF.IDF weighted vector space.[10] Concretely, we take the subset of maximally 25 terms

[9] The exported RDF triples are ingested in the “Verrijkt Koninkrijk” triple store. The updated triple store can be found at `http://semanticweb.cs.vu.nl/verrijktkoninkrijk/`.

[10] This is similar to how the widely used open-source search library Lucene implements so-called “more like this” similarity queries. We use their default settings of maximally 25 terms and at least 30% matching terms. See: `http://lucene.apache.org/core/5_4_1/queries/org/apache/lucene/queries/mlt/MoreLikeThis.html`

from a source document that have the highest TF.IDF score. We then select documents that match at least 30% of these terms and compute similarity as the sum of TF.IDF scores over the terms. More matching terms thus lead to a higher similarity, as does matching a less common term than a more common term.

We measure temporal similarity using a Gaussian decay function. If we compare two news articles, that both have a specific publication date, this temporal similarity functions exactly the same as a temporal document prior. If two documents are from the same date, they are completely temporally similar. The further the two documents are apart in time, the lower the similarity score. When either document does not have a specific publication date, we need to compute temporal similarity based on the automatically extracted temporal references. Concretely, we multiply the Gaussian decay score we obtain for each temporal reference match.[11] The overall similarity between two documents is then computed by multiplying the temporal similarity with the textual similarity. In this way, temporal similarity functions in a similar matter as a temporal document prior would work, giving preference to documents from a specific period.

3.2.5 Evaluating Related Article Finding

We evaluate our approach to measuring similarity using a retrieval experiment to find related documents within the Wikipedia collection. The task is to find documents related to a source Wikipedia article within the Wikipedia collection. We compare two approaches for finding related documents: using only textual similarity and combining temporal and textual similarity.

We sample ten Wikipedia articles out of the 18,361 articles that link to an article with WWII in the title (*"Tweede Wereldoorlog"* in Dutch). We pool the top ten results based on textual similarity and have annotators judge the relatedness of two documents side-by-side on a four-point scale, labeled from bad to perfect. We obtain the relatedness labels via a crowdsourcing platform. To ensure good quality judgments, we manually create a set of gold standard judgments for twelve document pairs that pilot judges agreed entirely on. Our crowdsourcing judges need to obtain an agreement of over 70% with the gold standard to start judging. During judging, gold standard pairs are intertwined with unjudged pairs. If the judges do not maintain this agreement on the gold standard, they cannot continue and their judgments are not included. We obtain at least three judgments per document pair, more if the three judges do not agree. We obtain 812 judgments (including the judgments for the gold standard) and measure a mean agreement of 69.5% over all document pairs. We compute a final relatedness rating for a document pair as the mean rating over all judges for that pair.

In our application, related documents are presented to find interesting alternative perspectives from different collections. The related documents are presented as a ranked list, very similar to a standard information retrieval setting. Given this setting and the annotations on a four-point scale, we choose nDCG@10 as our evaluation metric (see also the background on IR evaluation and nDCG in Section 2.1.3). The nDCG metric can incorporate graded relevance and gives more importance to results higher in the ranked

[11] Our temporal similarity approach is based on the decay function implementation in Elasticsearch (open-source and based on Lucene): `https://www.elastic.co/guide/en/elasticsearch/reference/2.2/query-dsl-function-score-query.html#function-decay`.

Table 3.3: Effectiveness of our approach combining textual and temporal similarity on the task of related article finding as measured by nDCG@10.

	nDCG@10
Only textual similarity	0.861
Combining textual and temporal similarity	0.894 (+3.8%)

list. We compute nDCG only on the top ten results, as we expect that lower documents are unlikely to be inspected by a user. An nDCG score of 1 indicates that documents are ranked perfectly in order of their relevance score and a score of 0 would mean that all retrieved documents have the lowest relevance score.

Table 3.3 shows the effectiveness of our approach. Using only textual similarity, we measure an nDCG value of 0.861 and an average rating in the top ten of 2.6 on a scale from 1 to 4. This suggests that the retrieved documents are already of good quality and ranked in a reasonable order. By combining textual and temporal similarity we improve the nDCG score with 3.8% to 0.894. A detailed look at each of the ten source documents shows improvements in nDCG scores up to 25%, but also decreased scores of up to 5%. The results suggest that we effectively retrieve related documents and that combining textual and temporal similarity improves effectiveness over only using textual similarity.

There are interesting challenges for future research in new approaches for incorporating temporal similarity. For example, the decreased scores suggest that it could be beneficiary to use temporal similarity based on temporal references only in specific cases (e.g., when a source document covers only a small timescale). Furthermore, intuitively, a match on year references is less strong a match than one on day references. One could therefore weight matches based on the least fine-grained granularity. We revisit the task of related article finding in Chapter 7, albeit in a very different setting. There, we generate queries to find related content for live television, based on the textual stream of subtitles. In Chapter 7, we will show limited impact of a temporal document prior to favor more recent content, and a substantial impact of using a decay on the importance of terms from the stream of subtitles.

3.3 Search Interface to Explore Perspectives

We described how we connect collections and their content in Section 3.2. To illustrate how our algorithms could be used to automatically create connections between digital and digitized collections, we now present a novel search and analysis application.[12] We provide new means of exploring and analyzing perspectives in these connected collections using two insightful visualizations of the connected collections: (1) an exploratory search interface that provides insight into the volume of data on a particular WWII-related topic and (2) an interactive interface in which a user can select two articles/pages for a detailed comparison of the content. Our architecture is modeled in Figure 3.4. At the core of

[12] The fully functional application can be accessed at `http://qhp.science.uva.nl`.

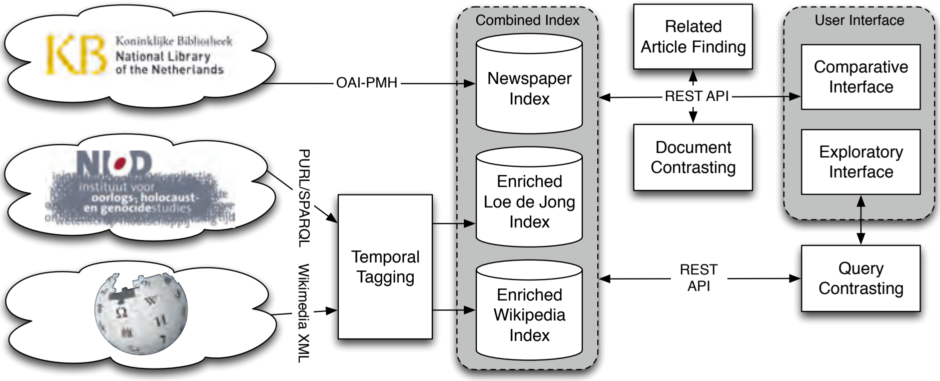


Figure 3.4: Architecture to support exploration of perspectives in collections. The interface (right) interacts with a combined index (center) collected from three sources (left).

the application we use proven open-source technology such as xTAS and Elasticsearch. For each collection, we build a separate index that is exposed to the user interfaces as a combined index.

A researcher interacts with our application through two separate, but connected interfaces: (1) an exploratory search interface for a broad overview, and (2) a comparative interface for detailed analysis. We will discuss the flow between these interfaces with a worked example of a researcher interacting with the application in Section 3.4. First, we will describe each of the two interfaces in more detail below. The comparative interface communicates directly with the combined index, supported by related article finding and document contrasting services. For the exploratory interface, all requests to the index are processed through the query contrasting system.

To explore perspectives in our application, we provide the research with means to do contrasting. In the comparative interface, two documents are contrasted in detail in a side-by-side comparison. In the exploratory interface, a researcher can combine a keyword query with predefined "query contrasts." A query contrast is as a set of filters that each define a collection subset. A single filter functions in a similar way as facet filters. Such a query contrast can simply be contrasting different collections (e.g., newspaper versus Wikipedia articles), or different sources (e.g., a collaborating newspaper versus one run by the resistance) or different locations of publishing. Using a set of these filters (what we call a *query contrast*), what is expressed as a simple keyword query turns into a contrasting comparison between different perspectives.

3.3.1 Exploratory Interface

To allow researchers to explore the three connected collections, we build an exploratory interface as part of our application. This interface is sketched in Figure 3.5.

Based on a keyword query and a contrast in the search bar on top, an overview of the search results is presented. Central in the exploratory interface is a visualization that shows the distribution of the volume of documents across time. We visualize this distribution as a streamgraph [41], that can be seen as a streamlined version of a stacked

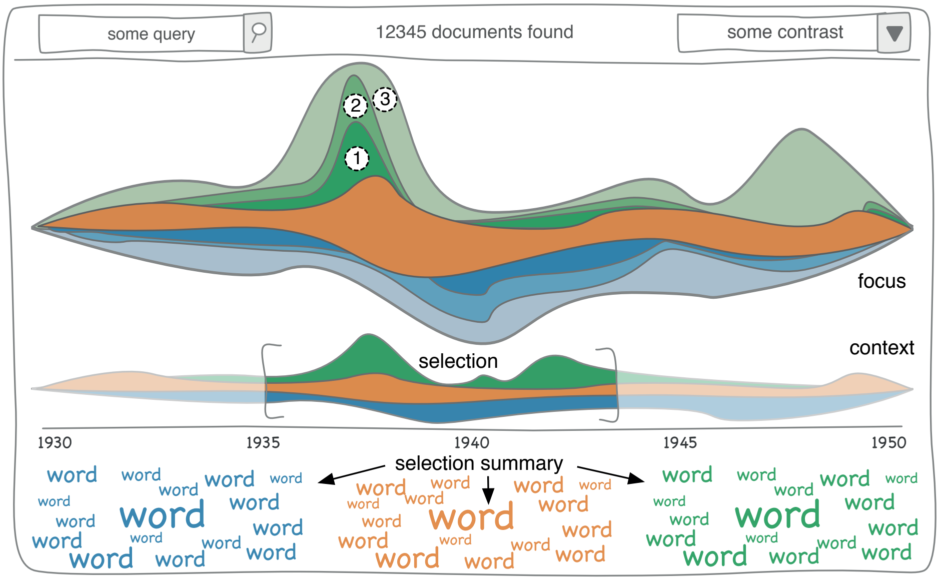


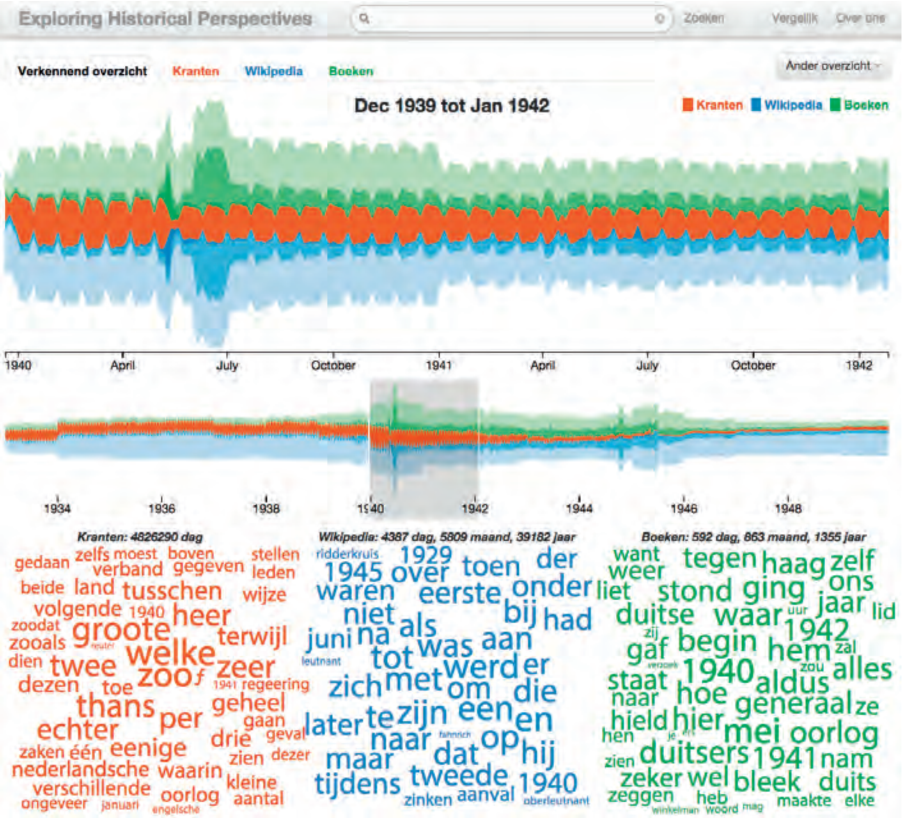
Figure 3.5: Sketch of the exploratory interface with search on top, a streamgraph to visualize document volume and word clouds to summarize each collection. Marked with a number are the substreams for different temporal granularity, as explained in the text.

bar chart. A researcher can select a time period of interest while maintaining an overview through a Focus+Context interaction design [43]. This allows researchers to focus on a specific period, while at the same time getting an impression of the entire time period.

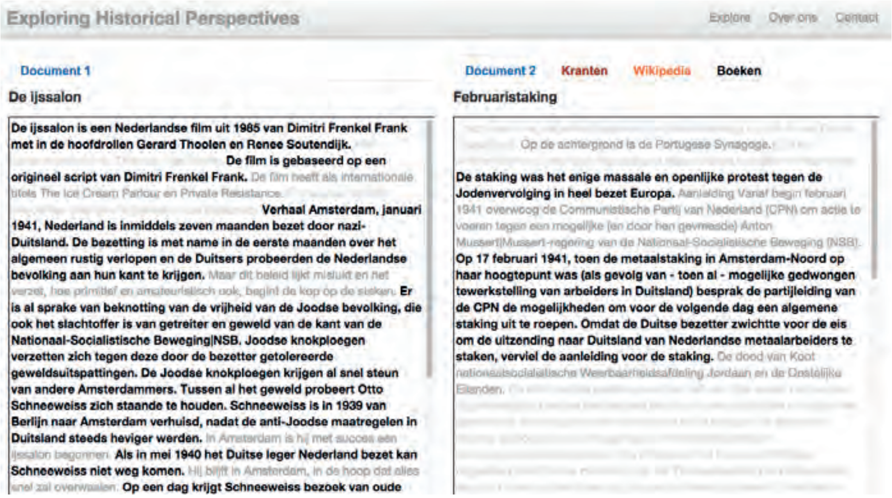
The streams in the context visualization are defined by the selected query contrast and consistently color coded based on this. In the simplest case, each represents one of the three connected collections: newspaper articles, encyclopedic articles and sections of the reference books. For each stream, we show a word cloud representing the most significant terms in the documents in each stream for the selected time period (based on TF.IDF). This provides the researcher with a quick overview of what topics these documents cover.

In Section 3.2.2, we described how we extract temporal references that can have a granularity of a day, month or even a year. If a stream is based on extracted date references, we distinguish between different temporal reference granularity in the focus stream graph. A stream is then split into three substreams: (1) day references, (2) month references and (3) year references (marked in Figure 3.5). The color coding is kept consistent, but opacity decreases as temporal references become less fine-grained. Similarly, the more fine-grained day references are positioned closer to the center of the stream. If a document refers to a specific day, it refers to that month and year as well. In the visualization, we do not count a reference for a less fine-grained substream if a more fine-grained reference occurs for that day. This way, the combined height of the three substreams at any point of time is equal to the references to that day, month or year.

A screenshot of the exploratory interface is shown in Figure 3.6a. Not depicted in Figure 3.6 is the collection search interface, that shows a simple ranked list of documents within any of the three collections. From this search interface, a researcher can select a


Exploring Historical Perspectives
Zoeken
Vergelijk
Over ons
Verkennend overzicht
Kranten
Wikipedia
Boeken
Ander overzicht
Dec 1939 tot Jan 1942
🟥 Kranten 🟦 Wikipedia 🟩 Boeken
1940
April
July
October
1941
April
July
October
1942
1934
1936
1938
1940
1942
1944
1946
1948
Kranten: 4826290 dag
Wikipedia: 4387 dag, 5809 maand, 39182 jaar
Boeken: 592 dag, 863 maand, 1355 jaar
gedaan zelfs moest boven stellen verband gegeven leden beide land tusschen wijze volgende 1940 heer zoodat groote terwijl zooals reuter dien welke twee zeer dezen toe zoo 1941 regeering thans geheel echter per gaan drie geval zaken één eenige zien dezer nederlandsche waarin kleine verschillende oorlog aantal ongeveer januari engelsche
ridderkruis 1929 toen der 1945 over waren eerste onder niet bij had juni na als was aan luitenant tot werd er zich met om die later te zijn een en naar fuhrrich op maar dat hij tijdens tweede 1940 zinken aanval oberleutnant
want tegen haag zelf weer liet stond ging ons jaar lid duitse waar uur 1942 gaf begin zij hem zal verzoek 1940 zou alles staat aldus naar hoe generaal ze hield hier je hen mei oorlog zien duitsers 1941 nam zeker wel bleek duits zeggen heb winkelman woord mag maakte elke

(a) Screenshot of exploratory interface (sketched in Figure 3.5).



De ijssalon

De ijssalon is een Nederlandse film uit 1985 van Dimitri Frenkel Frank met in de hoofdrollen Gerard Thoolen en Renee Soutendijk. **De film is gebaseerd op een origineel script van Dimitri Frenkel Frank.** De film heeft als internationale titels The Ice Cream Parlour en Private Resistance. **Verhaal Amsterdam, januari 1941, Nederland is inmiddels zeven maanden bezet door nazi-Duitsland. De bezetting is met name in de eerste maanden over het algemeen rustig verlopen en de Duitsers probeerden de Nederlandse bevolking aan hun kant te krijgen.** Maar dit beleid lijkt mislukt en het verzet, hoe primitief en amateuristisch ook, begint de kop op de steken. **Er is al sprake van beknotting van de vrijheid van de Joodse bevolking, die ook het slachtoffer is van getreiter en geweld van de kant van de Nationaal-Socialistische Beweging|NSB. Joodse knokploegen verzetten zich tegen deze door de bezetter getolereerde geweldsuitspattingen. De Joodse knokploegen krijgen al snel steun van andere Amsterdammers. Tussen al het geweld probeert Otto Schneeweiss zich staande te houden. Schneeweiss is in 1939 van Berlijn naar Amsterdam verhuisd, nadat de anti-Joodse maatregelen in Duitsland steeds heviger werden.** In Amsterdam is hij met succes een ijssalon begonnen. **Als in mei 1940 het Duitse leger Nederland bezet kan Schneeweiss niet weg komen.** Hij blijft in Amsterdam, in de hoop dat alles snel zal overwaaien. **Op een dag krijgt Schneeweiss bezoek van oude**

Februaristaking

Op de achtergrond is de Portugese Synagoge.

De staking was het enige massale en openlijke protest tegen de Jodenvervolging in heel bezet Europa. Aanleiding Vanaf begin februari 1941 overwoog de Communistische Partij van Nederland (CPN) om actie te voeren tegen een mogelijke (en door hen gevreesde) Anton Mussert|Mussert-regering van de Nationaal-Socialistische Beweging (NSB). **Op 17 februari 1941, toen de metaalstaking in Amsterdam-Noord op haar hoogtepunt was (als gevolg van - toen al - mogelijke gedwongen tewerkstelling van arbeiders in Duitsland) besprak de partijleiding van de CPN de mogelijkheden om voor de volgende dag een algemene staking uit te roepen. Omdat de Duitse bezetter zwichtte voor de eis om de uitzending naar Duitsland van Nederlandse metaalarbeiders te staken, verviel de aanleiding voor de staking.** De dood van Koot nationaalsocialistische Weerbaarheidsafdeling Jordaan en de Oostelijke Eilanden.

(b) Screenshot of comparative interface.

Figure 3.6: Screenshots of exploratory (a) and comparative (b) interfaces.

document to study in more detail in the comparative interface.

3.3.2 Comparative Interface

The comparative interface shows two documents side-by-side (see Figure 3.6b). At first, a selected document is shown on one side, while the other side shows related documents from each of the three collections using the approach described and evaluated in Section 3.2. When selecting a document from these results, the side-by-side comparison is shown. A researcher can return to the related articles on either side at any time.

When comparing two documents side-by-side, interesting parts of the document are highlighted. Using an approach similar to the textual similarity described in Section 3.2.4, we compute the similarity of each sentence in a document to the document on the other side. Sentences with a high similarity are shown clearly, whereas sentences with a low similarity are shown partially transparent. This dimming effect draws the attention of the researcher to the interesting parts of a document in the context of another.

3.4 A Worked Example

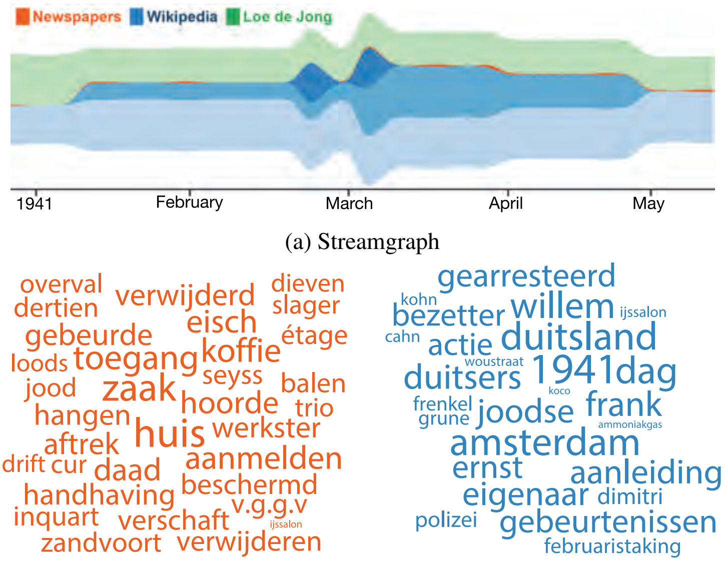
We describe a worked example of how our application can be used to study different perspectives on a specific event in WWII. We go back to an event in Amsterdam, early 1941, as described on Wikipedia. During the winter months of ’40/’41, oppression of Jewish citizens of Amsterdam was rising. This lead to open street fights between mobs of both sides. The tensions culminated on February 19th in an ice cream parlor called Koco, where a fight broke out between a German patrol and a mob of regular customers set out to defend the shop. Several arrests were made and the Jewish-German owners where arrested and deported. After roundups in other parts of Amsterdam, the tensions finally lead to the “February strike,” the only massive public protest against the persecution of Jews in occupied Europe.

Figure 3.7 shows screenshots of the exploratory search interface, when searching for the name of the shop, contrasting the three distinct collections. From the streamgraph, the historian can clearly see that most documents that mention the shop are from or refer to early 1941. Focusing on this period of interest, the streamgraph depicted in Figure 3.7a shows some references to the shop in the buildup towards this event with the bulk in early 1941. A detailed look at the word clouds for newspapers (Figure 3.7b), shows that emphasis is given to the perspective of the police. The significant terms include: robbery, thieves, enforcement, gain access, removed and case.[13] On the other hand, the Wikipedia articles referring to this event focus more on the human interest and broader perspectives. The word cloud in Figure 3.7c shows the names of the owners and terms as cause, events, February strike, arrested and owner.[14]

Diving deeper into the different perspectives, the historian searches for articles related to Section 8.2 of Loe de Jong’s fourth book, part II, that covers the events around Koco. He finds Wikipedia articles covering the February strike and related events, but decides to have a more detailed look at the article on the movie “The Ice Cream Parlour.” Figure 3.8

[13] In Dutch: *overval, dieven, handhaving, toegang, verschaft, verwijderd, zaak.*

[14] In Dutch: *aanleiding, gebeurtenissen, Februaristaking, gearresteerd, eigenaar.*



(a) Streamgraph

(b) Newspapers

(c) Wikipedia

Figure 3.7: Screenshots for query “ijssalon koco” from October 1941 until March 1942.

shows a screenshot of the comparison of the content of this article in comparison with the section written by Loe de Jong. The sentences that focus mostly on the movie are faded out, drawing attention to the parts of the article that describe the events in February 1941.

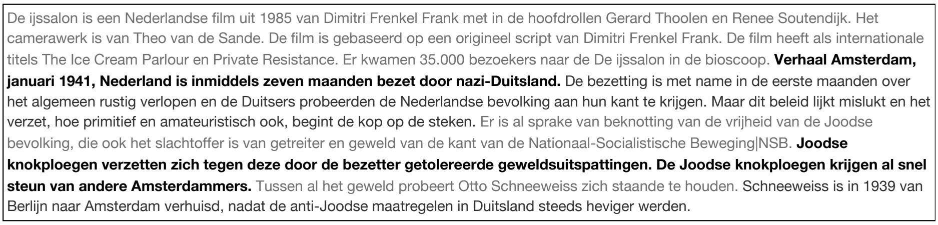
This worked example illustrates how each collection has different perspectives on an important event in WWII, both in comparing subsets of the collection (Figure 3.7) and in comparing two documents (Figure 3.8). One can easily think of follow-up questions for a historian to explore after this, for example: how does the perspective of newspapers from Amsterdam differ from those in the rest of the country? Our examples illustrates how the automatically created connections between collections benefits a historian in studying these news archives.

3.5 Conclusion

We started this chapter with a motivational use case on how historians study large news collections. We presented an approach to connect multiple heterogeneous collections through implicit events, via time and content. We have cast this as an IR task and have asked:

RQ1 Can we effectively extract temporal references from digitized and digital collection and does leveraging these temporal references improve the effectiveness of retrieving related articles?

In answer to RQ1, we found that we can extract temporal references effectively from digital and digitized collections. We found that digitized collections pose interesting chal-



De ijssalon is een Nederlandse film uit 1985 van Dimitri Frenkel Frank met in de hoofdrollen Gerard Thoolen en Renee Soutendijk. Het camerawerk is van Theo van de Sande. De film is gebaseerd op een origineel script van Dimitri Frenkel Frank. De film heeft als internationale titels The Ice Cream Parlour en Private Resistance. Er kwamen 35.000 bezoekers naar de De ijssalon in de bioscoop. **Verhaal Amsterdam, januari 1941, Nederland is inmiddels zeven maanden bezet door nazi-Duitsland.** De bezetting is met name in de eerste maanden over het algemeen rustig verlopen en de Duitsers probeerden de Nederlandse bevolking aan hun kant te krijgen. Maar dit beleid lijkt mislukt en het verzet, hoe primitief en amateuristisch ook, begint de kop op de steken. Er is al sprake van beknotting van de vrijheid van de Joodse bevolking, die ook het slachtoffer is van getreiter en geweld van de kant van de Nationaal-Socialistische Beweging|NSB. **Joodse knokploegen verzetten zich tegen deze door de bezetter getolereerde geweldsuitspattingen. De Joodse knokploegen krijgen al snel steun van andere Amsterdammers.** Tussen al het geweld probeert Otto Schneeweiss zich staande te houden. Schneeweiss is in 1939 van Berlijn naar Amsterdam verhuisd, nadat de anti-Joodse maatregelen in Duitsland steeds heviger werden.

Figure 3.8: Screenshot showing the first sentences of the Dutch Wikipedia article on the movie "The Ice Cream Parlour" compared to Loe de Jong's article on the events around that parlour.

lenges requiring improved preprocessing. Leveraging these extracted temporal references improved effectiveness on the task of retrieving related articles. We consider our proposed approach for using extracted temporal references to improve related article finding as just a first attempt. We have identified interesting challenges in extracting temporal references from historical narratives, such as the books of Loe de Jong. Improving the accuracy of temporal reference extraction on such a collection poses interesting future work for information extraction researchers. How should we model and combine partial temporal references? What distance between references should we consider when combining references: a paragraph, a page, half a book? Is this perhaps different for running text than for section headers or the title?

To illustrate how our algorithms could be used to automatically create connections between digital and digitized collections, we have introduced a novel search interface to explore and analyze the connected collections that highlights different perspectives and requires little domain knowledge. We showed the value of our application through a worked example of how to study different perspectives on an event in WWII. We have left an analysis of the commonalities between the interfaces and a full evaluation of how such interfaces would be used by researchers for future work. In Section 8.2, we will discuss several approaches for this evaluation, each with their own drawbacks and benefits.

While we focused in this work on events and collections related to WWII, our approaches can be applied to any kind of digital collections. We use two dimensions present in nearly every collection: time and textual content. We release our work as open data and open-source software[15] to foster future research, including on other collections. One can easily find examples in smaller timespans, e.g., investigating changing reputation of a company on Twitter, or finding when new words appear in a language, by analyzing books and news. In two demonstrators (not included in the thesis), we have considered tasks that are closely related to the work in this chapter: (1) to study the development of words over time [160] and (2) aimed at answering the question: "who put an issue on the agenda?" [57]. In both, we have considered only a single collection. The interesting question would be whether our approaches for connecting collections would work in such settings as well.

We will address some of the follow-up questions that arise from this chapter in the remainder of the thesis. Staying within the first research theme, in Chapter 4, we will study

[15] The source code is available on `https://bitbucket.org/qhp`.

framing, a concept from communication science that is related to the perspectives we have studied here. We focussed in this chapter on researchers that explore large collections. This often a very time-consuming activity [201]. From web search, we know that long search sessions occur when searchers are exploring, or when they are struggling to find relevant information [135, 228]. In the second research theme, in Chapter 5, we will return to the setting of long search sessions to study how web searchers behave when they cannot find what they are looking for. In the third and final research theme, we will return to the task of finding related content. There, we start from a live television broadcast and seek content related to the broadcast. In Chapter 6, we propose an approach for generating links to encyclopedic content related to live TV content. In Chapter 7, we study a task that is even more similar to the related article finding task described in Section 3.2.4. There, we generate queries to find related content for live television, based on the textual stream of subtitles and use reinforcement learning to directly optimize retrieval effectiveness.

4 Automatic Thematic Content Analysis: Finding Frames in News

We continue our research within the first research theme by turning our attention to the social sciences. In this chapter, we propose an automatic thematic content analysis approach based on how researchers study framing in news. In communication science, framing is the way in which journalists depict an issue in terms of a ‘central organizing idea’ [76]. Frames can be seen as a perspective on an issue, albeit in a more subtle sense than the perspectives that we offered historians in the previous chapter. There, a perspective was grounded in the source of an article (e.g., comparing a collaborating newspaper with one from the resistance). Here, these perspectives arise from an emphasis in how a journalist frames the news story.

In social science research, there is a growing trend of applying computational thinking and computational linguistic approaches. In particular, information retrieval (IR) is proving to be a useful but underutilized approach that may be able to make significant contributions to research in a wide range of social science domains [47]. One particular domain in which this is happening is the study of news and its impact. Early examples focus mostly on analyzing factual aspects in news, such as the impact of news on corporate reputation [144]. Increasingly, however, we are also seeing the use of IR to analyze more subjective aspects of news in social science research [123]. Similarly, IR research is broadening, from a strong focus on analyzing facts to also include more subjective aspects of language, such as opinions and sentiment [171], human values [70], argumentation [170] and user experiences from online forums [107].

In the social sciences, mass communication (such as news) is often studied through a methodology called *content analysis* and evolves around the question: “Who says what, to whom, why, to what extent and with what effect?” [121]. The aim of content analysis is to systematically quantify specified characteristics of messages. When these characteristics are complex, *thematic content analysis* [202] can be applied: first, texts are annotated for indicator questions (e.g., “Does the item refer to winners and losers?”) and the answers to such questions are subsequently aggregated to support a more complex judgment about the text (e.g., the presence of a conflict frame). Content analysis is a laborious process, and there is a clear need for a computational approach. Such an approach can improve the consistency, efficiency, reliability and replicability of the analyses, as larger volumes of news can be studied in a reproducible manner. In this chapter, we operationalize frame analysis as a classification task and ask the following research question:

RQ2 Can we approach human performance on the task of frame detection in newspaper articles by following the way-of-working of media analysts?

Following the way-of-working of media analysts, we propose a two-stage approach, where we first rate a news article using indicator questions for a frame and then use the outcomes to predict whether a frame is present.

The remainder of this chapter is organized as follows: in Section 4.1 we discuss media frame analysis; Section 4.2 describes our proposed methods and Section 4.3 describes the experimental setup. We present and discuss our results in Section 4.4, after which we conclude in Section 4.5.

4.1 Media Frame Analysis

News coverage can be approached as an accumulation of "interpretative packages" in which journalists depict an issue in terms of a *frame*, i.e., a central organizing idea [76]. Frames are the dependent variable when studying the process of how frames emerge (*frame building*) and the independent variable when studying effects of frames on predispositions of the public (*frame setting*) [193]. When studying the adoption of frames in the news, content analysis of news media is the most dominant research technique. Using questions as indicators of news frames in manual content analysis is the most widely used approach to manually detecting frames in text. Indicator questions are added to a codebook and answered by human coders while reading the text unit to be analyzed [202]. Each question is designed such that it captures the semantics of a given frame. Generally, several questions are combined as indicators for the same frame. This way of making inferences from texts is also referred to as thematic content analysis [183]. As described in Section 2.2.4, automatic or semi-automatic frame detection is rare. The approaches that do exist follow a dictionary-based [186] or rule-based [198] approach.

In this chapter, we focus on four commonly used generic news frames, originally proposed by Semetko and Valkenburg [197]: the conflict frame, human interest frame, economic consequence frame and morality frame. They are listed in Section 4.3, together with their indicator questions.

The *conflict frame* highlights conflict between individuals, groups or institutions. Prior research has shown that the depiction of conflict is common in political news coverage [154], and that it has inherent news value [74, 218].

By emphasizing individual examples in the illustration of issues, the *human interest frame* adds a human face to news coverage. According to Iyengar [102], news coverage can be framed in a thematic manner, taking a macro perspective, or in an episodic manner, focusing on the role of the individual concerned by an issue. Such use of exemplars in news coverage connects to research on personalization of political news [102].

The *economic consequence frame* approaches an event in terms of its economic impact on individuals, groups, countries or institutions. Covering an event with respect to its consequences is argued to possess high news value and to increase the pertinence of the event among the audience [75].

The *morality frame* puts moral prescriptions or moral tenets central when discussing an issue or event. Morality as a news frame has been studied in various academic publications and is found to be applied in the context of various issues such as, for example, gay rights [157] and biotechnology [32].

In this chapter, we contribute over and above the related work discussed here and in Section 2.2.4, by presenting and evaluating an ensemble-based classification approach for frame detection in news. To the best of our knowledge, this is the first work in which text classification methods are applied to this central issue in studying media. Furthermore, we investigate whether explicitly modeling the thematic content analysis approach improves performance.

4.2 Frame Classification

We approach the task of frame detection in news as a classification task. As described in Section 2.1.2, in a text classification task, the aim is to assign a particular label to a document based on the textual content. The assumption underlying thematic content analysis is that frames manifest themselves in a news article in a manner that is measured using indicator questions. We follow this assumption and analyze the wording in a news article in order to make a decision about the presence of frames.

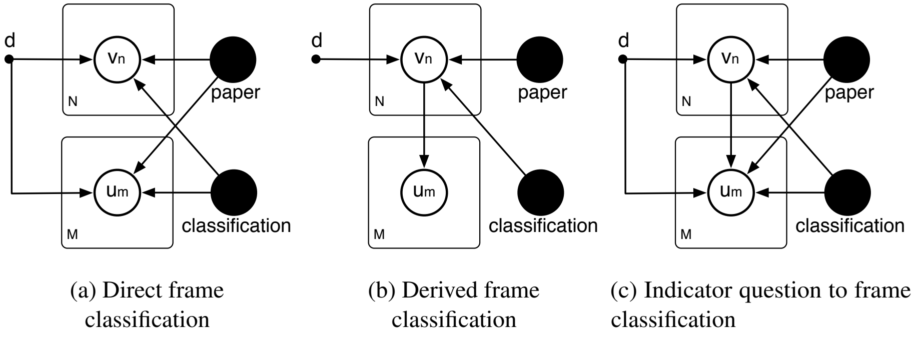
Given a collection of documents D and a set of frames U for which a set of indicator questions V have been defined, we estimate the probability $P(u_m|d)$ that a frame $u_m \in U$ is present in document $d \in D$. In thematic content analysis this probability is deconstructed into $P(u_m|v_1, \ldots, v_N)$ and a set of probabilities for each $v_n \in V$: $P(v_n|d)$. This formal definition of the task can be used for the automatic classification and for manual content analysis. In the latter case, the probability $P(v_n|d)$ is estimated using manual coding by humans after reading the document.

Document representation We represent documents as a bag-of-words with TF.IDF scores for each word. We apply sublinear term frequency scaling, i.e., replace the term frequency (TF) with $1 + \log(TF)$, use l2 normalization and smooth IDF weights by adding one to document frequencies. We have evaluated other representation (e.g., n-grams and topic models), but these did not improve classification performance and will not be reported here.

Besides the words represented in the document, we extend the document representation with information on the source of the document and with a classification for each document (i.e., a topic, such as finance, infrastructure, etc.). The extended bag-of-words document representation serves as the features for classification.

Frame and indicator question classification We propose two baselines and three approaches for automatic indicator question and frame classification. These methods differ in how the coherence between indicator question and frame is modeled. The approaches are depicted in graphical models in Figure 4.1 and will be described below.

Stratified random classification baseline Our first baseline approach is very naive and intended as a lower bound. It randomly choses the answer to a indicator question or whether a frame is present or not, taking into account only the prevalence in the training set. This naive baseline randomly assigns a classification, without considering the document



(a) Direct frame classification

(b) Derived frame classification

(c) Indicator question to frame classification

Figure 4.1: Graphical models of the three classification approaches. The circles represent random variables, where the filled are observable. The rectangular plates indicate multiple of these variables.

and its representation, with a probability based on the class distributions. This naive baseline will be more likely to randomly pick the majority class than the minority class.

Direct classification baseline Our second baseline approach is to classify answers to indicator questions and the presence of each frame directly. More formally, we train a classifier to estimate $P(u_m|d)$ for each frame $u_m \in U$. This approach is the simplest approach and is depicted in Figure 4.1a. Note that for frames, we completely ignore the indicator questions in this baseline approach.

For classification we use Logistic Regression to optimize logistic loss using Pegasos-style regularization [199]. For training we alternate between pairwise ROC-optimization and standard stochastic gradient steps on single examples [196]. This baseline approach aims to be flexible in dealing with issues such as class imbalance.

Ensemble-based direct classification Our first approach is to improve binary classification decisions for indicator questions and for the presence of a frame by using an ensemble of binary-class linear classifiers (also depicted in Figure 4.1a). The predictions of all these classifiers are the features for a final classifier. The ensemble includes different linear support vector machines (SVMs), linear rank-based SVMs [109, 196], and Perceptron-based algorithms [117]. This ensemble-based approach aims to be flexible in dealing with the different complex characteristic of each of the classifications. We combine the classifiers in the ensemble using the same classifier as described above for the direct classification baseline approach (Logistic Regression using Pegasos regularization).

Derived frame classification Our second approach is to infer the presence or absence of a frame based on the classification for indicator questions. More formally, we train an ensemble-based classifier to estimate $P(\hat{v}_n|d)$ for each indicator question $v_n \in V$. We then derive the probability $P(u_m|d)$ for each frame $u_m \in U$ from $P(\hat{u}_m|\hat{v}_1, \ldots, \hat{v}_N)$ for all indicator questions $v_m \in V$. This approach is depicted in Figute 4.1b and closely resembles the manual approach, where human coders make binary decisions for $P(v_n|d)$ for each $v_n \in V$ and $d \in D$.

Indicator question to frame classification Our third approach is a cascade approach, where we first classify for the indicator question and then use the outcomes to classify the frames. More formally, we train an ensemble-based classifier to estimate $P(\hat{v}_n|d)$ for each indicator question $v_n \in V$. We then train an ensemble-based classifier to estimate the probability $P(u_m|d, \hat{v}_1, \ldots, \hat{v}_N)$ for each frame $u_m \in U$. This approach is depicted in Figure 4.1c. Practically, we implement this by adding ensemble-based predictions for indicator questions as features for the frames classifiers.

4.3 Experimental Setup

To evaluate our methods we run a number of experiments. We describe the document collection used, outline how the four frames have been coded in the manual content analysis that we use as training and test data, and explain how we evaluate the performance of our classification models.

Document collection Our document collection consists of digital versions of front page news articles of three Dutch national daily newspapers (*De Volkskrant*, *NRC Handelsblad* and *De Telegraaf*) for the period between 1995 and 2011. These articles come from the Dutch Lexis-Nexis newspaper archive, and each article has a topical classification (based, e.g., on the location in the newspaper). The used sample is a stratified sample of 13% for each year.

Indicator Questions Annotations For each year covered in our collection, a random sample of news articles was taken. This sample was filtered (based on manually assigned labels) to only contain articles that were political in nature. The resulting 5,875 documents have been manually coded for the presence of four generic news frames (described in Section 4.1). Indicator questions were used to code the news frames.

A total of thirteen yes-or-no-questions were used as indicators of the news frames. In previous research, these questions have been shown to be reliable indicators of the four frames [197]. The indicator questions for each frame are:

C *Conflict frame*:

- **C1** Does the item reflect disagreement between parties, individuals, groups or countries?
- **C2** Does the item refer to winners and losers?
- **C3** Does the item refer to two sides or more than two sides of the problem?

E *Economic consequence frame*:

- **E1** Is there a reference to the financial costs/degree of expense involved, or to financial losses or gains, now or in the future?
- **E2** Is there a reference to the non-financial costs/degree of expense involved, or to non-financial losses or gains, now or in the future?
- **E3** Is there a reference to economic consequences of pursuing or not pursuing a course of action?

H *Human interest frame*:

- **H1** Does the item provide a human example or human face on the issue?
- **H2** Does the item employ adjectives or personal vignettes that generate feelings of outrage, empathy caring?

- **H3** Does the item mention how individuals and groups are affected by the issue or problem?
- **H4** Does the item go into the private or personal lives of the actors?

M *Morality frame*:

- **M1** Does the item contain any moral message?
- **M2** Does the item make reference to morality, God or other religious tenets?
- **M3** Does the item offer specific social prescriptions about how to behave?

Manual coding was conducted by a total of 30 trained coders. All coders were communication science students and native speakers of the Dutch language. In order to assess inter-coder reliability, a random subset of 159 articles was coded by multiple coders. Measures of the percentage of inter-coder agreement range from 70% to 94%. The inter-coder reliability is included in the results in Table 4.1 and Table 4.3, with the label 'Human.'

Frame Annotations Based on the annotations for indicator questions, a second annotation round gave rise to the construction of frame annotations, following the methodology described by Semetko and Valkenburg [197]. To establish the coherence of the indicator questions and their relation to the frames a factor analysis is performed. We find a four factor solution for the answers to the indicator questions. In this solution each indicator question has a loading onto each factor (i.e., a weight).

In these factor loadings, we can identify the four frames, i.e., for each frame there is a factor with high loads for the corresponding indicator questions and low loadings for the others. For two indicator questions (C2 and E2) the factor load is below 0.5, and hence these were considered unreliable indicators (in line with [197]). This means that the remaining indicator questions can be considered reliable indicators of the four frames: a frame is considered present in a news document whenever any of the indicator questions corresponding to the frame is answered positively.

Evaluation metrics We perform ten-fold cross-validation and compare the agreement between human annotators and our automatic approach in terms of agreement. Where possible, we evaluate both the answers to indicator questions and the frame annotations. Furthermore, we compare the approaches in receiver operating characteristics (ROC) space. We compare the ability to distinguish true positive classifications from false positives for different operating characteristics that will produce increasingly more positive results. In this ROC space, we can compute the area under the curve (AUC). The AUC metric for a classifier expresses the probability that the classifier will rank a positive document above a negative document.

4.4 Results and Discussion

Table 4.1 and Table 4.3 describe the agreement between our approaches and the human annotations for each of the eleven indicator questions and the four frames. For comparison, these tables also include the inter-annotator agreement for human coders. Table 4.2 and Table 4.4 describe the area under the curve (AUC) metric for our approaches.

Table 4.1: Agreement between automatic classification predictions and human annotations for each of the eleven indicator questions and the three approaches (two baselines and ensemble).

	C1	C3	E1	E3	H1	H2	H3	H4	M1	M2	M3
Random	0.5214	0.5980	0.7093	0.8419	0.7963	0.8346	0.5144	0.9122	0.9348	0.9397	0.9535
Direct	0.5709	0.6140	0.7093	0.8419	0.7963	0.8346	0.5750	0.9122	0.9348	0.9397	0.9535
Ensemble	0.7064	0.6945	0.8511	0.8650	0.8007	0.8393	0.6489	0.9137	0.9345	0.9460	0.9535
Coder biased ensemble run is included below for analysis.											
Biased	0.7200	0.7413	0.8553	0.8819	0.8213	0.8494	0.7045	0.9185	0.9346	0.9501	0.9525
Human inter-coder agreement is included below for comparison. Note that this is evaluated on a small dataset.											
Human	0.7239	0.6994	0.8282	0.8466	0.7546	0.7055	0.6748	0.8405	0.9080	0.9041	0.9202

Table 4.2: Area under the curve (AUC) for ROC of automatic classification predictions compared to human annotations for each of the eleven indicator questions and the two direct approaches (baseline and ensemble).

	C1	C3	E1	E3	H1	H2	H3	H4	M1	M2	M3
Direct	0.6235	0.6601	0.6973	0.6885	0.6283	0.5802	0.6027	0.5960	0.5572	0.6591	0.4903
Ensemble	0.7744	0.7672	0.8966	0.8432	0.7483	0.7419	0.7051	0.7990	0.6917	0.8884	0.6509

Table 4.3: Agreement between automatic classification predictions and human annotations for each of the four frames and the five frame classification approaches. Significant differences, tested using a two-tailed Fisher randomization test against the "Direct" approach, are indicated for the other automatic classification approaches with ▲ ($p < 0.01$).

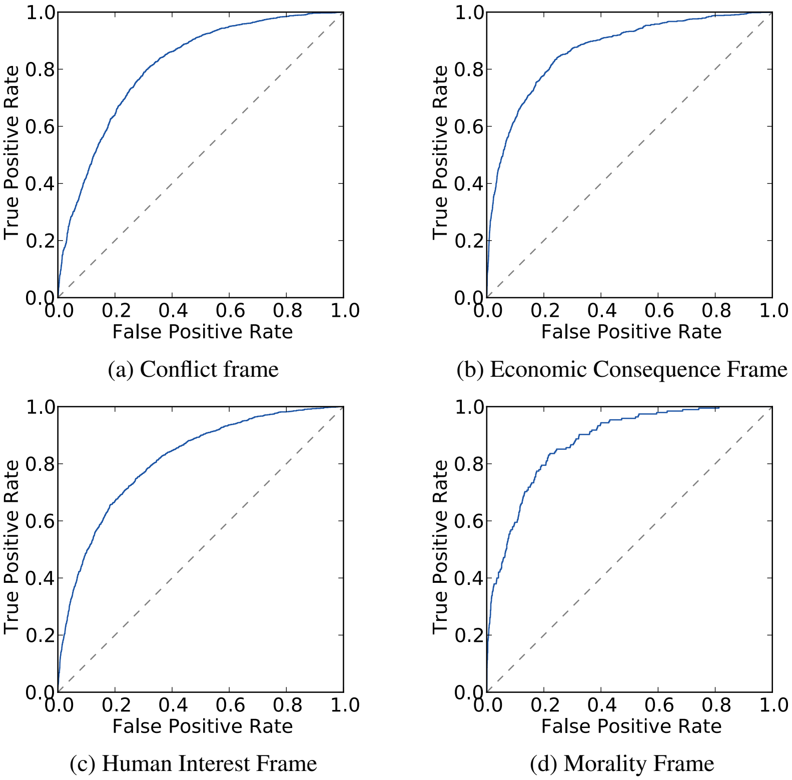
	C	E	H	M
Random	0.6403	0.5755	0.6231	0.8679
Direct	0.6654	0.8134	0.7779	0.9668
Ensemble	0.7241▲	0.8506▲	0.7949▲	0.9668
Derived	0.5709▼	0.7093▼	0.6158▼	0.9348▼
IQ → F	0.7202▲	0.8489▲	0.8014▲	0.9677
Coder biased ensemble run is included below for analysis.				
Biased	0.7501▲	0.8642▲	0.8141▲	0.9685
Human agreement on small dataset included for comparison.				
Human	0.7730	0.8160	0.6442	0.8528

Table 4.4: Area under the curve (AUC) for ROC of automatic classification predictions compared to human annotations for each of the four frames and four frame classification approaches.

	C	E	H	M
Direct	0.6379	0.6956	0.6008	0.5909
Ensemble	0.7802	0.8496	0.7580	0.7597
Derived	0.5575	0.5000	0.5897	0.5000
IQ → F	0.7677	0.8436	0.7748	0.8025

Indicator Questions Classification Results We can observe in Table 4.1 that our baseline single classifier direct approach ("Direct") performs well on some of the indicator questions, but worse on others. The direct baseline is unable to consistently improve over the naive stratified random baseline ("Random"). Our ensemble-based approach ("Ensemble") substantially improves over these baselines and achieves accuracy scores ranging from 65% accuracy upwards. While we observe that the accuracy varies among the four frames and the corresponding indicator questions, our ensemble-based approach is able to capture the complex characteristics for all questions and frames. The conflict indicator questions (C1 and C3) and human interest question H3 perform below average in the baselines, but perform substantially better in the ensemble-based approach.

Human interest question H4 and the morality questions (M1, M2 and M3) show high baseline performance, but do not show substantially improvements for the direct approaches, despite our pairwise optimization approach. This suggests that these questions are underrepresented and possibly less well represented using a bag-of-words approach than the other questions.



(a) Conflict frame

(b) Economic Consequence Frame

(c) Human Interest Frame

(d) Morality Frame

Figure 4.2: ROC Curves for the ensemble-based direct approach for the four frames.

Looking at the AUC metric results in Table 4.2, we see the same substantial improvements of the ensemble-based approach over the direct classification baseline. We can also observe a substantial improvement for the aforementioned indicator questions H4, M1, M2 and M3. This suggests that while we are not better in terms of accuracy for these questions, we are indeed better at estimating the probability of a document belonging to a class.

Frame Classification Results We can observe in Table 4.3 that accuracy scores on frames follow the same pattern as the indicator questions. The conflict and human interest frame prediction again performs worse than the others. Interestingly, we can observe a substantial improvement for the morality frame over the stratified random baseline. The ensemble-based approach is able to obtain significant improvements over the baselines approaches. We can also observe that deriving the scores from the indicator questions does not perform well, directly predicting scores for frames using the ensemble-based approach performs substantially better. Interestingly, the two-stage indicator question to frame classification approach does not perform better than the direct approach. The additional information we add by first classifying the indicator questions does not help in classifying the frames. The results for the AUC metric (described in Table 4.4) show a

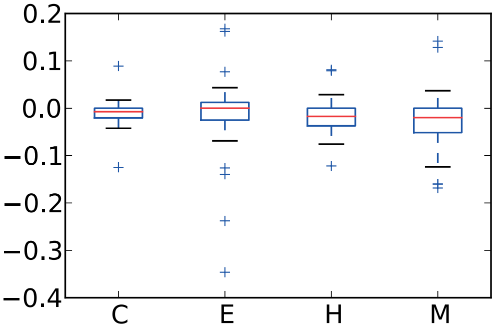


Figure 4.3: Box plot of the weights on the binary coder variables for the four frames in one of the ensemble SVM classifiers.

qualitatively similar pattern as the agreement.

Furthermore, we can observe from Table 4.1 and Table 4.3 that the morality frame and the corresponding questions perform strikingly well in all approaches in terms of agreement. A plausible explanation for this is that this frame is a lot less prevalent than the other three (present in 13% of the documents, compared to 64% for conflict, 58% for economic consequence and 62% for human interest). The AUC results in Table 4.2 and Table 4.4 provide some evidence that these classifiers still perform up to par.

To validate this, and to obtain more insight into the operation characteristics of the classifiers we take a more detailed look at the ROC curves. Figure 4.2 shows these ROC curves for the direct ensemble-based approach for the frames. We can observe a similar curve for each of the frames. From these graphs and the AUC results, we can conclude that while we can not perfectly classify the frame annotations, we are able to obtain a good rate of true positives if we allow some false positives.

Human Inter-Coder Agreement Compared to human inter-coder agreement, nearly all accuracy scores for the ensemble-based and two-stage approaches are at or above that level. Note, however, that human agreement is evaluated on a much smaller dataset. We observe a lower performance compared to human agreement for question H3, the conflict frame and corresponding questions C1 and C3. For the morality frame and the human interest and morality questions the human inter-coder agreement is even below the stratified random baseline.

To investigate the difficulty of each question and the quality of the human annotations, we look at whether the annotations for questions are stable across coders. We measure this by evaluating a new ensemble-based model where the document representation is extended with variables representing the coder. This creates the unrealistic but insightful scenario where we predict the answer of a specific coder to a specific question. This model allows us to compensate for a bias a coder might have, possibly resulting in higher performance compared to the regular ensemble-based approach.

Agreement for the biased model is included in Table 4.1 and Table 4.3. We observe increased performance for most questions, with C3, E3 and H3 standing out. For frames the performance is increased for the human interest frame, economic consequence frame and most substantially for the conflict frame.

To further investigate this, we look at the weights that the coder features get assigned in the biased model. If all coders would answer the indicator questions exactly the same, the coder features will have a weight very close to zero. A weight that differs from zero suggests a consistent difference in answers from one coder compared to the other coders. Figure 4.3 shows these weights for each frame in one of the classifiers in the direct frame ensemble classifier. We see that the weights do indeed deviate from zero, with a different range per frame. The economic consequence frame has the highest range, with a maximum of 0.5 bias per coder on a scale of -1 to 1. These weights suggest consistently different interpretations of the indicator questions across coders.

4.5 Conclusion

In this chapter, we have proposed algorithmic approaches to finding frames in news that follow the manual thematic content analysis approach. Answering RQ2, our results provide strong evidence that we are able to approach human performance on predicting both the answers to indicator questions as well as the presence of a frame.

Our ensemble-based approach to directly predicting the presence of a frame is the most effective and improves substantially over the baseline approach. The derived approach, which mirrors the manual approach, was the least effective. Surprisingly, the more informed indicator question to frame classification approach did not perform better than the ensemble-based direct classification approach. This suggests that for the task of frame classification, explicitly modeling the manual thematic content analysis does not improve performance. Our ensemble-based direct classification approach is sufficient to capture the complex characteristics of frames that the indicator questions are aimed to represent.

The results of an analysis using a model that explicitly models coder bias and the relatively low inter-coder agreement suggest that coders have different interpretations of the indicator questions for the frames. Like the indicator questions that represent different aspects of complex characteristics of messages, it seems that human coders represent different views on these aspects and characteristics.

For the task of frame detection in news, we have shown that, using an ensemble-based classification approach (trained on previously labelled documents), we are able to approach human performance in terms of accuracy on this task. A combined approach of human and automated frame detection seems to be the logical way forward. In such an active learning scenario, the selection of documents to annotate next is informed by the current classification model. An active learning approach could, for example, select the document for annotation that it is most uncertain about. An interesting research question would be whether using active learning would lead to reduced annotation effort for achieving the same accuracy. Furthermore, existing approaches for active learning typically consider a single classification task, whereas in the frame detection task we consider multiple frames per document. When selecting documents for annotation, would it be better to consider the current prediction for only a single frame or combine predictions for multiple frames and how would one combine this?

Our approach for automatic frame detection in news has broader applications. In Chapter 3, we presented an interactive search application for historians that highlights different perspectives in large collections. There, we considered clearly different perspectives on events in WWII, e.g., the reporting in a newspaper collaborating with the occupiers versus an illegal newspaper of the resistance. When studying less controversial news topics, automatic frame detection could provide an approach for finding different perspectives that is grounded in communication science. A researcher could be suggested an article that covers the same event as the one he is currently reading, but with different framing, e.g., stressing the impact on an individual instead of a conflict between parties. Such a related article is likely to provide an interesting additional perspective and also insight into how the journalists view the shared topic.

In a more generic search setting, automatic frame detection can be used to provide searchers with more diverse search results on controversial topics. For example, Yom-Tov et al. [232] algorithmically encourage people to read diverse political opinions by increasing their exposure to varied political opinions, with the goal of improving civil discourse. For this, they consider web search queries that are clearly associated with a certain political stance (e.g., *obamacare* or *tea party*). During ten days, the Microsoft Bing web search engine showed merged search results for this query with search results for a matching query from the opposing side. They find that people who were shown more diverse results continued reading more diverse news and overall became more interested in news. A similar effect could exist for news that is framed differently and our algorithmic approach could support this.

Part II

Struggles and Successes

5 Struggling and Success in Web Search

The second part of the thesis is motivated by a mixed-methods study on how web searchers behave when they cannot find what they are looking for. Later in this part, within research theme 3, we will propose to help searchers using two methods to automatically retrieve content based on subtitles. First, this chapter covers the second research theme of this thesis. We turn our attention to the domain of web search and a long session scenario.

When searchers experience difficulty in finding information, their struggle may be apparent in search behaviors such as issuing numerous search queries or visiting many results within a search session [11]. Such long sessions are prevalent and time-consuming (e.g., around half of Web search sessions contain multiple queries [201]). Long sessions occur when searchers are exploring or learning a new area, or when they are struggling to find relevant information [135, 228]. Methods have recently been developed to distinguish between struggling and exploring in long sessions using only behavioral signals [92]. However, little attention has been paid to *how* and *why* searchers struggle. This is particularly important since struggling is prevalent in long tasks, e.g., Hassan et al. [92] found that in 60% of long sessions, searchers' actions suggested that they were struggling.

Struggling leads to frustrating and dissatisfying search experiences, even if searchers ultimately meet their search objectives. Better understanding of search tasks where people struggle is important in improving search systems. We address this important issue using a mixed methods study using large-scale logs, crowd-sourced labeling, and predictive modeling and ask two research questions:

RQ3 How do web searchers behave when they cannot find what they are looking for?

RQ4 How do web searchers go from struggle to success and how can we help them make this transition?

Before proceeding, let us present an example of a struggling task. Figure 5.1 presents a session in which a searcher is interested in watching live streaming video of the U.S. Open golf tournament. The first two queries yield generic results about U.S. Open sporting events and the specific tournament. The third query might have provided the correct results but it included the previous year rather than the current year. At this stage, the searcher appears to be struggling. The fourth query is the so-called *pivotal query* where the searcher drops the year and adds the terms "watch" and "streaming." This decision to add these terms alters the course of the search task and leads to a seemingly successful

```
9:13:11 AM  Query  us open
9:13:24 AM  Query  us open golf
9:13:36 AM  Query  us open golf 2013 live
9:13:59 AM  Query  watch us open live streaming
9:14:02 AM  Click  http://inquisitr.com/1300340/watch-2014-u-s-open-
                           live-online-final-round-free-streaming-video
9:31:55 AM  END
```

Figure 5.1: Example of a *struggling* task from June 2014.

outcome (a click on a page that serves the streaming content sought). Understanding transitions between queries in a struggling session (e.g., the addition of "golf" and the wrong year), and transitions between struggling and successful queries (e.g., the addition of terminology pertaining to the desired action and content), can inform the development of strategies and algorithms to help reduce struggling.

Related research has targeted key aspects of the search process such as satisfaction, frustration, and search success, using a variety of experimental methods, including laboratory studies [11, 65], search log analysis [89], in-situ explicit feedback from searchers [72], and crowd-sourced games [2]. Such studies are valuable in understanding these important concepts, and yield insights that can directly improve search systems and their evaluation. However, they do not offer insights into how people who initially struggle, in some cases, ultimately succeed. Such insights can have value to both searchers and search providers in detecting problems and designing systems that mitigate them. We address these shortcomings with the research described in this chapter. We make the following specific contributions:

- We use large-scale search log analysis to characterize aspects of struggling search tasks and to understand how some tasks result in success, while others result in failure.
- We propose and apply a crowd-sourced labeling methodology to better understand the nature of the struggling process (beyond the behavioral signals present in log data), focusing on why searchers struggled and where it became clear that their search task would succeed (i.e., the pivotal query).
- We develop a classifier to predict query reformulation strategies during struggling search tasks. We show that we can accurately classify query reformulations according to an intent-based schema that can help select among different system actions. We also show that we can accurately identify pivotal queries within search tasks in which searchers are struggling.
- We propose some application scenarios in which such a classifier, and insights from our characterization of struggling more broadly, could help searchers struggle less.

The remainder of this chapter is structured as follows. Section 5.1 briefly recaps the related work in areas such as satisfaction, success, and query reformulation. Section 5.2 characterizes struggling based on our analysis of large-scale search log data. In Section 5.3, we describe our crowd-sourced annotation experiment and the results of the analysis. Section 5.4 describes predictive models and associated experiments (namely, predicting query transitions and identifying the pivotal query), and the evaluation results. In Section 5.5 we discuss our findings, their implications and limitations, and conclude.

5.1 Related Work

Characterizing the behavior of searchers has been a subject of study from different perspectives using a range experimental methods. Of particular interest to our research is the extensive body of work, which we discussed in detail above, on (1) satisfaction and search success (§2.3.1), (2) searcher frustration and difficulty (§2.3.2), and (3) query reformulation and refinement (§2.3.3).

The concepts of satisfaction and search success discussed in (§2.3.1) are related, but not equivalent. Success is a measure of goal completion and searchers can complete their goals even when they are struggling to meet them [89]. Satisfaction is a more general term that not only takes goal completion into consideration, but also effort and more subjective aspects of the search experience such as searcher's prior expectation [105]. Related to satisfaction are other key aspects of the search process such as task difficulty and searcher frustration (discussed in §2.3.2). These have been studied using a variety of experimental methods, including log analysis [92], laboratory studies [11, 65], and crowd-sourced games [2]. When searchers experience difficulty in finding information, their struggle may be apparent in search behaviors such as issuing numerous search queries, more diverse queries or visiting many results within a search session [11]. However, rather than struggling, these longer sessions can be indicative of searchers exploring and learning [62, 228]. Hassan et al. [92] have recently developed methods to distinguish between struggling and exploring in long sessions using only behavioral signals. They found that searchers struggled in 60% of long sessions.

In this chapter, we focus on struggling tasks (that are thus likely to be unsatisfactory) to understand how some of them end up successful while others end up unsuccessful. We target traditional Web search in this study given its prevalence. For more detailed insight on searcher behavior, we analyze query reformulations (discussed in (§2.3.3). Query reformulation is the act of modifying the previous query in a session (adding, removing, or replacing search terms) with the objective of obtaining a new set of results [90]. For this, a number of related taxonomies have been proposed [8, 84, 98, 122, 209]. Huang and Efthimiadis [98] surveyed query reformulation taxonomies and provided a mapping between these and their own approach. While they provide interesting insight into searchers trying to articulate their information needs, these approaches all focus on superficial lexical aspects of reformulations.

No previous study examines how searchers struggle and what makes them ultimately succeed. We employ large-scale log analysis and a crowd-sourced labeling methodology to provide new insight into the nature of struggling and what contributes to their success. Based on this, we propose a new taxonomy for intent-based query reformulation that goes beyond the superficial lexical analysis commonly applied in the analysis of query transitions (see Huang and Efthimiadis [98]). Building on our characterization, we propose predictive models of key aspects of the struggling process, such as the nature of observed query reformulations and the prediction of the pivotal query. Based on insights gleaned from our data analysis, we also provide some examples of the types of support that search systems could offer to help reduce struggling.

5.2 Characterizing Struggling

We apply large-scale search behavioral analysis to characterize aspects of struggling search tasks and to understand how some of these searches end up successful while others end up unsuccessful.

5.2.1 Definitions

We focus on a particular type of search task that exhibits search behavior suggestive of struggling. We assume a broad view of struggling behavior and apply the following definitions:

Struggling describes a situation whereby a searcher experiences difficulty in finding the information that they seek. Note that in this definition they may or may not eventually locate the target of their search [92].

Sessions are a sequence of search interactions demarcated based on a 30-minute user inactivity timeout [58, 226].

Tasks are defined as topically-coherent sub-sessions, i.e., sequences of search activity within sessions that share a common subject area [92]. We follow the approach of [92] and assume that two queries belong to the same task if they are less than ten minutes apart and the queries match one of the following conditions: (1) share at least one non-stop word term, or (2) share at least one top ten search result or domain name (where popular domains such as wikipedia.org are excluded).

Struggling tasks describe topically coherent sub-sessions in which searchers cannot immediately find sought information.

Quick-back clicks describe result clicks with a dwell time (i.e., the time spent on a landing page after a click) of less than ten seconds [114].

Since we cannot infer that searchers experience difficulty from a single query, we only consider longer struggling tasks in our analysis. Our aim is to obtain a broad understanding of struggling search behavior in natural settings. To do this, we study search tasks where struggling is very apparent. Intuitively, when a searcher cannot locate the information they are seeking, they are much less likely to click search results and examine landing pages. We focus on tasks where a searcher does not examine any of the search results for the first two queries in detail. This includes queries that do not receive any clicks as well as queries with clicks that result in only very short dwell time on the landing page (quick-back clicks).

5.2.2 Mining Struggling Tasks

To better understand struggling search behavior in a natural setting, we analyze millions of search sessions from the Microsoft Bing Web search engine. We select these sessions from the interaction log of Bing for the first seven days of June 2014. All logged interaction data (i.e., queries and clicked search results) are grouped based on a unique user identifier and segmented into sessions. We mine struggling tasks from all sessions using the following steps:

1. **Filter sessions:** We included only search sessions originating from the United States English language locale (en-US), and excluded internal traffic and secure traffic (https). Furthermore, we only considered sessions that started with a typed query (and not with a click on a query suggestion).
2. **Segment sessions into tasks:** Using time-based segmentation can lead to combining multiple unrelated tasks into a single session. We therefore further refine sessions into topically coherent sub-sessions that cover a single task.
3. **Filter struggling tasks:** From these tasks, we select those with at least three queries where the first two lead to either no clicks or only quick-back clicks.
4. **Partition based on final click:** Lastly, we partition the struggling tasks based on searcher interaction for the last query in the task. Although we cannot directly infer whether the searcher successfully fulfilled their information need, search activity for terminal queries has been shown to be a reasonable proxy for success [72, 89]. We validate this predictor using crowd-sourced annotations in Section 5.3. More specifically, we partition the tasks into three sets:
   (a) **Unsuccessful:** If a searcher does not click on any result for the final query or when their only clicks are quick-back clicks (less than 10 seconds), we assume the searcher was unsuccessful and abandoned their search task without satisfying their need.
   (b) **Successful:** If a searcher clicks on a search result and we observe no interaction for at least 30 seconds, we assume the searcher was successful. Note that this includes tasks in which the searcher does not return at all to the result page before the session times out.
   (c) **Other:** All other tasks have clicks where searchers examine landing pages between 10 and 30 seconds. Based on previous research [72], we consider these task outcomes to be ambiguous and exclude them.

Following these steps, we obtain two sets of tasks that differ only based on the interaction with the terminal query. The combined dataset contains nearly 7.5 million struggling tasks. We now describe characteristics of these tasks to provide insight into struggling behavior, seeking to understand how some searches result in success, while others are unsuccessful. We begin with struggling tasks, and then consider queries and query reformulations.

5.2.3 Task Characteristics

Of the struggling tasks in our set, approximately 40% are successful per our definition. Around half of the tasks comprised three queries and successful tasks were slightly shorter than their unsuccessful counterparts. Focusing on the relationship between task duration and task outcome, Figure 5.2 shows the percentage of struggling tasks that were continued after a specified number of queries. We observe from the figure that the percentage of

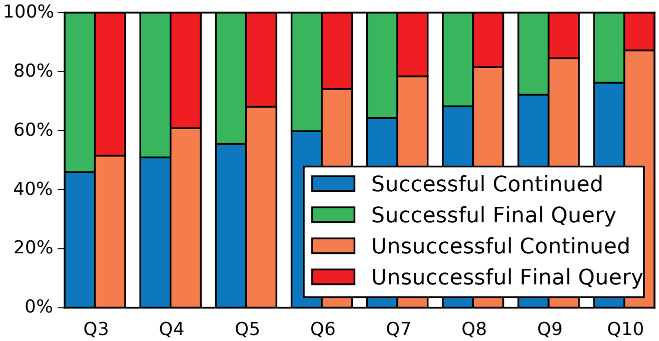


Figure 5.2: Percentage of successful and unsuccessful struggling tasks continued (onto another query) or completed (task terminates) after the third query (Q3) until the tenth query (Q10).

tasks that are continued increases with the number of queries already issued (i.e., the likelihood of a re-query increases with each successive query). Unsuccessful tasks are continued more frequently than successful tasks. There are many important factors, such as searcher tenacity and sunk cost, that may explain more of a reluctance to abandon unsuccessful tasks.

For some topics, searchers experience more difficulty in finding what they are looking for than for others. For example, Hassan et al. [92] reported that exploratory behavior is more than twice as likely as struggling behavior when shopping for clothing compared to downloading software. To obtain greater insight in the relationship between search topics on struggling behavior, we analyze the top 10 search results returned to searchers. We classify each document in the search results using the top-level categories of the Open Directory Project (ODP). For this we use an automatic content-based classifier [21]. We assign the most frequently-occurring topic across all queries in a task as the topic of that task. Figure 5.3 shows the prevalence of topics in struggling tasks. The proportion of successful tasks ranges from 47% in Computers to 27% in Arts, which is also the most prevalent topic. Next, we analyze the changes within a task by considering characteristics at different stages.

5.2.4 Query Characteristics

Since we have tasks of different lengths, we can only directly compare tasks of the same length or align queries in tasks of different length. To analyze the development over the broadest set of tasks, we consider the first and last query in the search task, and collapse all intermediate queries.

Query length. We observe that the first query in both successful and unsuccessful tasks is typically short (3.35 and 3.23 terms respectively on average). Intermediate queries are typically longer, averaging 4.29 and 4.04 terms respectively. The final queries are also longer, averaging 4.29 and 3.93 terms respectively. Increased success associated with longer queries has motivated methods to encourage searchers to issue queries with more terms [1].

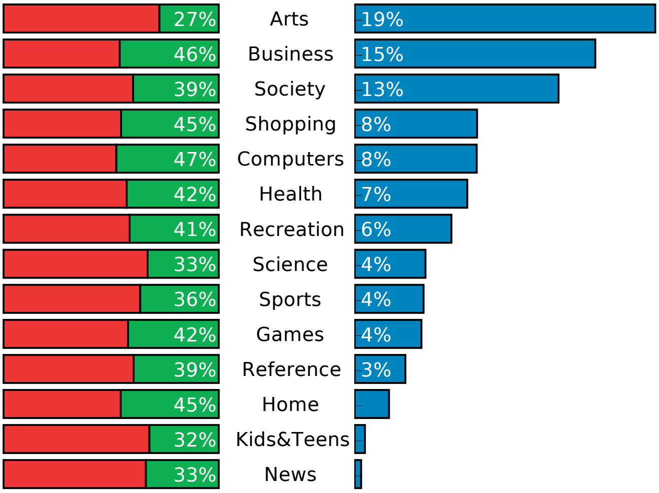


Figure 5.3: Prevalence of top-level ODP categories in all analyzed tasks (right) and proportion of successful and unsuccessful struggling tasks for that category (left).

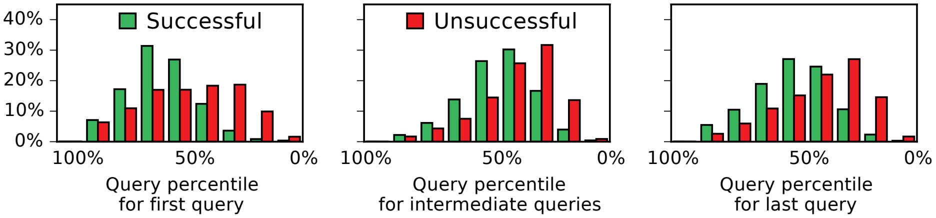


Figure 5.4: Query percentile computed over the frequency per query in the previous month. A larger percentile corresponds to a more frequent query.

Query frequency. For every queries in our dataset we compute the frequency of the query during the previous month. Figure 5.4 shows the percentile of a query, based on how often the query was issued in the previous month. Queries tend to become less common as the task progresses (distributions shift to the right in each plot). The first query is different from the ones that follow. For successful outcomes, the first query is more common than the successive queries, and more common than the first query for unsuccessful outcomes. This could suggest two different causes for struggling on the first query: (1) if it is common, it may be general and ambiguous, and (2) if it is uncommon, it might be overly specified.

5.2.5 Interaction Characteristics

Next, we turn to the interaction within a task. We follow a similar query grouping approach as in Section 5.2.4, using the position in the task. Since we select struggling tasks based on limited interaction on the first two queries, we separate those from the others. We now have three groups: first and second query, last query and all other intermediate queries.

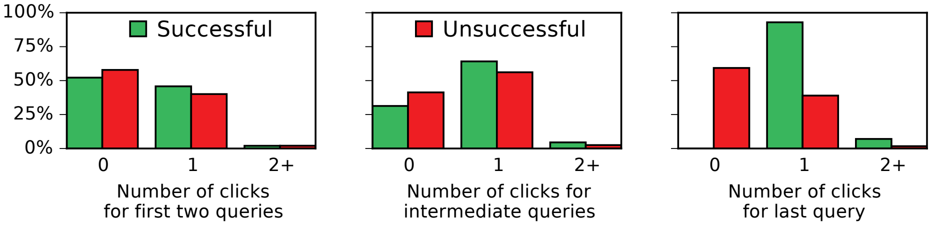
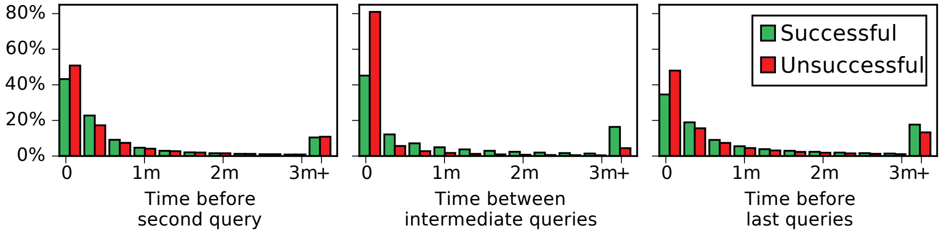
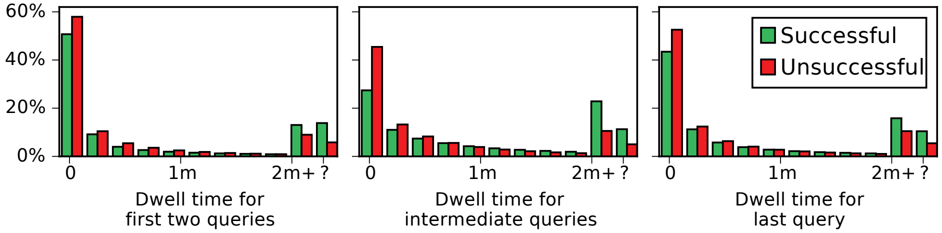


Figure 5.5: Number of result clicks at different queries.



(a) Time between queries



(b) Average dwell time (i.e., the time spent on a page after a click)

Figure 5.6: Time between queries and spent on a clicked page.

Note that we have no intermediate queries for tasks of only three queries (about half of the tasks). Figure 5.5 shows the change in the number of clicks in successful and unsuccessful tasks. We observe that for the first and second query, just over 40% of the tasks have a quick-back click. By our definition of struggling task (see Section 5.2), all clicks for the first two queries were quick-back clicks.

As described in Section 5.2, we used the clicks on the search results of the final query to partition our dataset into successful and unsuccessful tasks. We observe in Figure 5.5 that indeed the final query for all successful tasks has at least one click. The characteristics for the final query in the unsuccessful tasks is very similar to the first two queries (that are selected in the same way). When comparing the successful and unsuccessful tasks up to the final query, we observe that queries without clicks are more common in unsuccessful tasks.

Figure 5.6a shows the time between queries and Figure 5.6b shows the dwell time. Considering the successful tasks, we see the time between queries increases as the task progresses, mostly due to an increase in dwell time. Interestingly, the pattern for unsuccessful struggling tasks is different. We observe less time between queries mid-task perhaps due to fewer clicks (Figure 5.5) and more quick-back clicks.

5.2.6 Query Reformulation

To better understand the change in queries within a task, we analyze how the text of a query is refined over time. We classify each query reformulation into one of six types, described in Table 5.1. We use the algorithm developed by Hassan [88], which builds an automatic classifier for a subset of the query reformulation types presented by Huang and Efthimiadis [98].

Figure 5.7 shows the distribution over query reformulation types. The most common type of query reformulation is specialization, i.e., adding a term. This is most likely to occur directly after the first query, when it accounts for almost half of all query reformulations in successful tasks and a bit less in unsuccessful tasks. Substitutions and generalizations (removing a term) are the next most common reformulations. These query reformulation types are substantially less likely for the first query reformulation and more likely in the middle. Around 10% of the query reformulations are spelling corrections and an equal percentage entirely new queries. Spelling reformulations are most likely directly after the first query. New queries are most likely as second query and, surprisingly, as the last query. Others have observed similar patterns, e.g., Aula et al. [11] showed that searchers try many distinct queries toward the end of difficult search tasks. Lastly, revisiting a prior query is rare and is more likely in the middle of the task than at the final query.

One could argue that of these query reformulation types a specialization is most informative, since it defines an information need in greater detail. This is the most common type and substantially more common in successful tasks (39% of reformulations) than in unsuccessful tasks (30%). For these unsuccessful tasks, almost all other types of query reformulation are more common than in successful task. With more substitutions, spelling corrections, completely new queries and returning to previous queries, it appears that searchers in the unsuccessful tasks experience more difficulty selecting the correct query vocabulary.

Inspired by Lau and Horvitz [122], we examined the temporal dynamics of the query reformulations in addition to their nature. Figure 5.8 shows the likelihood of observing a particular reformulation type given the time that has elapsed since the previous query. If a

Table 5.1: Lexical-based query reformulation types.

Type	**Description**
New	Shares no terms with previous queries.
Back	Exact repeat of a previous query in the task.
Spelling	Changed the spelling, e.g., fixing a typo.
Substitution	Replaced a single term with another term.
Generalization	Removed a term from the query.
Specialization	Added a term to the query.

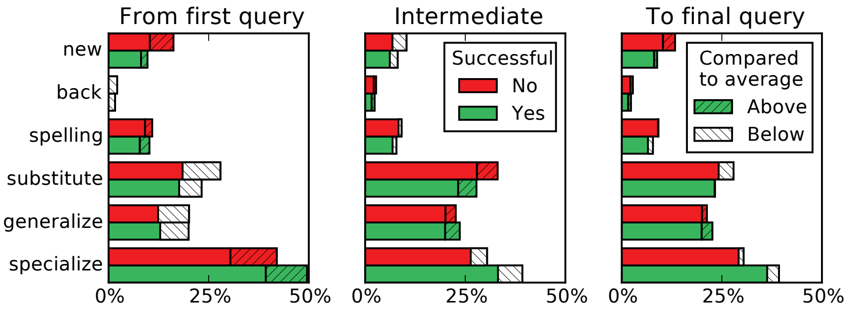


Figure 5.7: Distribution of query reformulation types at different stages of the search task for successful/unsuccessful tasks.

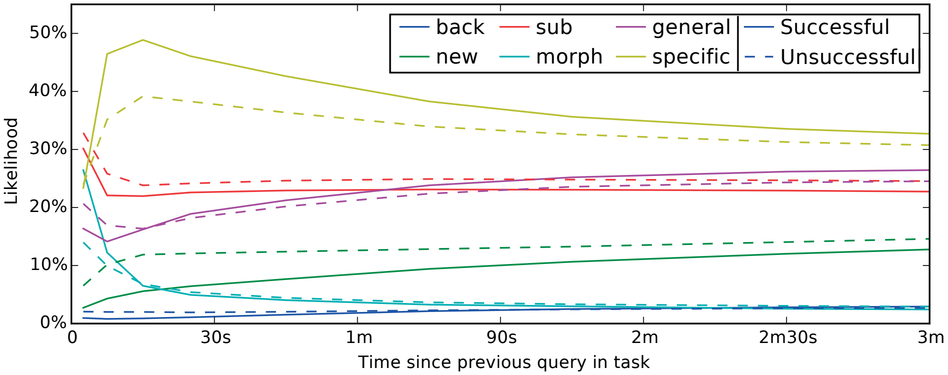


Figure 5.8: Likelihood of observing a query reformulation type in successful/unsuccessful tasks, given time since last query.

new query is issued within seconds after the previous, it is most likely to be a substitution or a spelling change. In fact, a spelling change is unlikely to occur after more than about fifteen seconds.

After a few seconds, the most likely type of reformulation is a specialization, with a peak at around fifteen seconds, where nearly half of the queries are expected to be reformulations. After this, some of the likelihood mass is taken over by generalizations and completely new queries. The likelihood per query reformulation type for successful vs. unsuccessful tasks appears similar, except for an increased likelihood of new queries for unsuccessful tasks, accompanied by a decreased likelihood of specialization. Anchoring on previous queries can harm retrieval performance [24]. The increase in new queries may represent an attempt to reset the query stream for the current task. Temporal dynamics such as these are interesting and may prove to be useful signals for the reformulation strategy prediction task described later in the chapter.

5.2.7 Summary

In the analysis in this section, we have shown there are significant differences in how struggling searchers behave given different outcomes. These differences encompass many aspects of the search process, including queries, query reformulations, result click behavior, landing page dwell time, and the nature of the search topic. Given these intriguing differences, we employ a crowd-sourcing methodology to better understand struggling search tasks and the connection between struggling and task outcomes.

5.3 Crowd-sourced Annotations

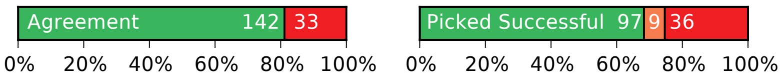
We performed a series of detailed crowd-sourced annotations to obtain a better understanding of what struggling searchers experience. This is also important in validating the assumptions made in the log-based estimation of search success from the previous section. Informed by an initial exploratory pilot with open-ended questions on a task level, we annotate tasks in detail on query transitions.

To obtain a representative and interesting sample of tasks (and importantly, to also control for task effects), we group the tasks described in Section 5.2 based on the first query in the task. Recall that these tasks are all struggling tasks, either successful or unsuccessful. For each initial query, we count the number of successful and unsuccessful tasks. We then filter these queries to have an approximately equal number of successful and unsuccessful tasks (between 45% and 55%).

Upon inspection of the selected tasks, we noticed a small set were unsuitable for annotation. To this end, we exclude initial queries that were deemed too ambiguous, are navigational in nature, or show many off-topic follow up queries. We randomly sample from these initial queries and verify whether they meet our criteria until we manually selected 35 initial queries for deeper analysis in an exploratory pilot study.

5.3.1 Exploratory Pilot

For each of the 35 initial queries, we generate five task pairs, by randomly sampling a successful task and an unsuccessful task that both start with that initial query. These 175 task pairs are annotated for struggling and success by three judges. We recruited a total of 88 judges from Clickworker.com, a crowd-sourcing service providing access to human annotators under contract to Microsoft and others. We created an interactive judgment interface that displays the search activity (queries and clicks) associated with two paired tasks side-by-side, in random order. To validate whether the activity on the terminal query is a reasonable proxy for success, we asked judges in which of the two tasks they believe the searcher was more successful using three options (one of the two sessions was more successful or they where indistinguishable). We have agreement by majority vote (i.e., at least two of the three judges agreed) for 81% of the pairs (Figure 5.9a). If there is a majority, the successful task is picked for 68.4% of the pairs and the unsuccessful struggling task for 25.4% of the pairs, while in 6.3% of the cases it is agreed that the tasks are indistinguishable (Figure 5.9b). These findings suggest that the judgment task is tractable and that tasks we labeled as successful automatically are indeed more often deemed successful by our judges.



(a) Agreement

(b) Distinguishable

Figure 5.9: Agreement (a) and distribution (b) of task outcomes of the distinguishable (majority agreement) tasks.

We first considered the task in which the judge believed the searcher was more successful. We informed the judge that we believed that the searcher started off struggling with their search, and asked them to look in detail at the task and answer a number of open-ended questions, starting with:

- Could you describe how the user in this session eventually succeeded in becoming successful? *Describe what made the user pull through. Note that this could be multiple things, for example: both fixing a typo and adding a specific location.*

We then considered the task in which the searcher was less successful and asked the judge to look in detail at that task and answer:

- Why was the user in this session struggling to find the information they were looking for? *Describe what you think made the user have trouble locating the information they were looking for. Note that this could be multiple things, for example: looking for a hard to find piece of information and not knowing the right words to describe what they are looking for.*
- What might you do differently if you were the user in this session? *Describe how you would have proceeded to locate the information they were looking for. Note that this could be multiple things, for example: using different terms (please specify) and eventually ask a friend for help.*

The answers to the questions provided diverse explanations for why searchers were struggling and what (could have) made them pull through. The answer typically described specific changes from one query to the other. Some of the explanations were topic-specific (e.g., suggesting a particular synonym for a query term) and some were quite generic (e.g., a suggestion to expand an acronym). We observed from these answers that the main reasons and remedies for struggling are not very well captured by the query reformulation types that are typically used, including the ones we described in Table 5.1. We adapted our main annotation efforts accordingly.

5.3.2 Annotations

The exploratory pilot suggested that third-party judges agree on labeling the success or failure of search tasks (especially in the positive case, where it may be more clear that searchers have met their information goals), and provided diverse clues on how and why searchers are struggling in a task. We now dive deeper into specific queries and what searchers did to remedy struggling.

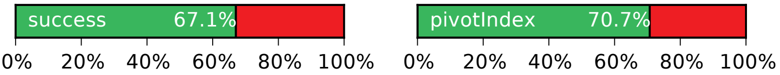
Table 5.2: Intent-based taxonomy presented to judges for each query transition, multiple response options could be selected.

Added, removed or substituted	□ an **action** (e.g., download, contact)
	□ an **attribute** (e.g., printable, free,), specifically (if applicable):
	  □ a **location** (or destination or route)
	  □ a **time** (e.g., today, 2014, recent)
	  □ **demographics** (e.g., male, toddlers)

Specified	□ a **particular instance** (e.g., added a brand name or version number)
Rephrased	□ Corrected a spelling error or **typo** □ Used a **synonym** or related term
Switched	□ to a **related** task (changed main focus) □ to a **new** task

In our main annotation efforts, we consider how a searcher reformulates queries within a task. Based on the open question answers of the exploratory annotation pilot, we propose a new taxonomy of query reformulation strategies (depicted in Table 5.2). In contrast to the lexical reformulation taxonomy in Table 5.1, this new taxonomy captures the intent of a query reformulation. Rather than simply observing that a term was substituted, we want to know if this is a related term to the one replaced or if it is a specification of an instance (e.g., refining [microsoft windows] to [windows 8]).

We hypothesize that there is a point during a struggling search task where searchers switch from struggling with little progress to making progress toward task completion (i.e., the so-called *pivotal* query). This could be associated with many factors, including the receipt of new information from an external source such as a search result or snippet. Understanding the pivotal query can provide valuable insight into what enabled the searcher pull through and is an important starting point when finding means to support searchers who are struggling. We ask judges to select the point in the task where they believe the searcher switched from struggling to being successful in finding what they were looking for. Judges could select either a query or a clicked URL in a task presented as shown in Figure 5.1. Judges could reissue the query to Bing and inspect the clicked pages that the original (logged) searcher selected. Furthermore, the crowd-workers were asked to judge how successful they think the searcher was in the task on a four-point scale: *not at all, somewhat, mostly, completely*. For annotation, we selected two separate, but closely-related sets of tasks. The first is based on the dataset from the exploratory pilot. We exclude six initial queries from the set that showed low agreement on picking the most successful task. For each of the 29 remaining initial queries, we sampled ten new successful tasks that started with that query. In a second set, we sampled 369 successful tasks with any initial query (as a control). This results in a total of 659 tasks.



(a) Success

(b) Pivotal Query

Figure 5.10: Levels of majority agreement for (a) four-level success and (b) on what query in the task was deemed to be pivotal.

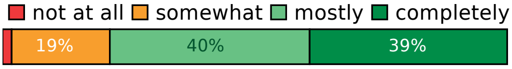


Figure 5.11: Judgments for the success of tasks.

5.3.3 Annotation Results

Each task was judged by three human annotators via an interactive interface. This results in a total of 1,977 annotations for 659 tasks. We removed the annotations of three of the 111 judges (80 annotations in total), because their responses were unrealistically quick. On average, judges spent 37 seconds to familiarize with a task and to judge the success and pick the pivotal query. Subsequently, they spent twelve seconds on average to judge each query transitions (at least two per task, depending on the number of queries).

Judges show 67% majority agreement on the task of judging the success of a searcher on a four-point scale (Figure 5.10a). Krippendorff's α measures 0.33, signaling fair agreement [9] between three judges. For judging what query is the pivotal query, we observe a 71% majority agreement for choosing one out of three or more queries ($\alpha = 0.44$, signaling moderate agreement between three judges, see also Figure 5.10b). This demonstrate that third-party labeling is feasible but also challenging given the subjectivity of the labeling task. We deem this satisfactory for our purposes and consider only annotations with majority agreement in the remainder of this section.

Figure 5.11 illustrates the distribution of success judgments. We selected these tasks assuming that the searcher was struggling, but eventually to some degree successful. We observe that in these tasks searchers are deemed to be at least mostly successful (79%), with only eight tasks (1.8%) not successful at all. This suggests that almost all studied searchers make at least some progress toward completing the task, even though they struggled along the way. We asked judges to consider the tasks carefully and select the pivotal query where searchers appeared to switch from struggling to succeeding. Figure 5.12 shows the distribution of positions of this pivotal query across the search task. In 62% of tasks, the pivotal query represents the final query in the task.

Query transitions. Judges were asked to examine each sequential pair of queries, while added and removed terms were highlighted. We asked judges to characterize the transition between these two queries by selecting one or more applicable options from the taxonomy. Figure 5.13 shows the distribution of query reformulation types. The most common are adding, substituting or removing an attribute and specifying an instance. The most common subtype of attribute modified is location with 17.8%, whereas time is only 5% of the attributes and demographic information (e.g., gender, age) occurs in

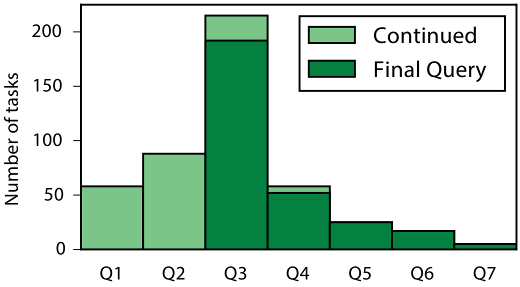


Figure 5.12: Pivotal query within a task, i.e., where the searcher appeared to switch from struggling to making progress.

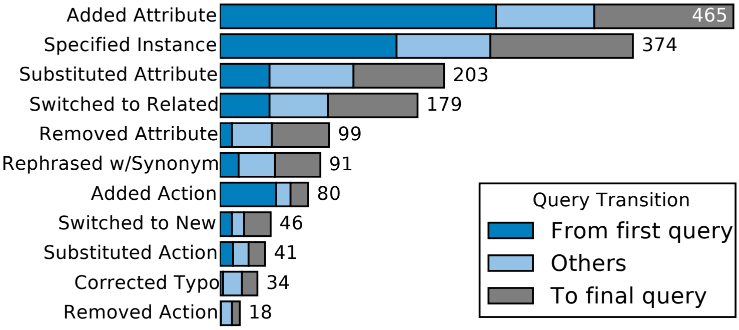


Figure 5.13: Distribution of query reformulation types.

only nine transitions (1.2%). None of the more fine-grained attribute subtypes show qualitatively different characteristics, so we will discuss only the attribute category, without the subtypes. Turning our attention to the different stages within tasks, we observe that adding attributes or actions and specifying an instance is relatively common from the first to the second query. For the transition towards the final query in a task, substituting or removing an attribute, rephrasing with a synonym and switching task are relatively common. This suggests more emphasis on broadly specifying information needs at the outset of tasks and more emphasis on refining it by adding specific details towards the end.

Transitions and task outcomes. Figure 5.14 shows the relationship between query transitions and success. It is worth noting that switching to a new task occurs substantially more often in less successful tasks. Specifying an instance also occurs relatively often in the less successful tasks. Addition, substitution, and deletion actions or attributions typically occurs in more successful tasks. Figure 5.14 also shows how often a query transition is deemed pivotal if it occurs. Interestingly, both switching tasks and specifying an instance are more common in less successful tasks, but are relatively frequently considered to be pivotal. Substituting an action is most often seen as pivotal, whereas substituting an attribute and correcting a typo are least frequently pivotal. The differences in the prevalence of pivotal queries as a function of the reformulation type suggests that

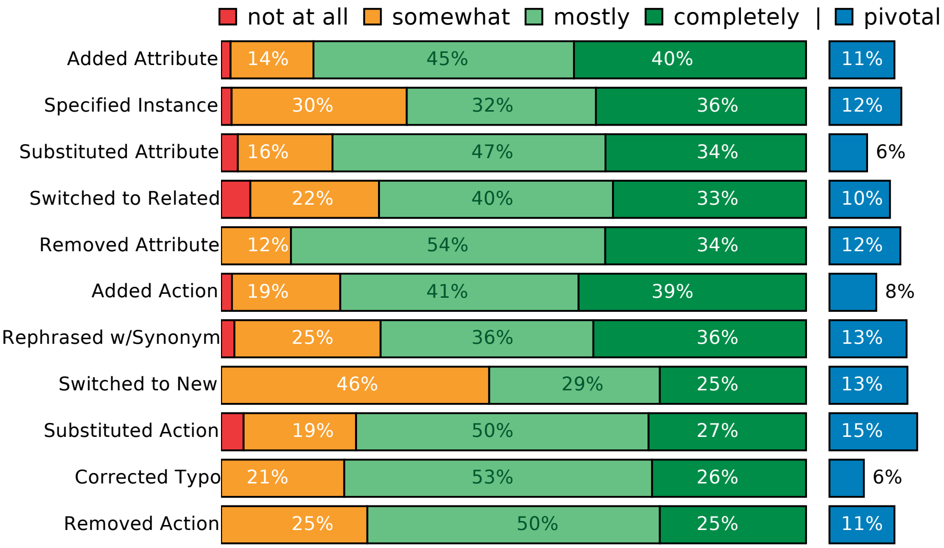


Figure 5.14: Task-level success on a four-point scale per reformulation type and the percentage of reformulations considered pivotal.

some actions may be more effective than others and that accurately predicting the next action presents the opportunity for early corrective intervention by search systems. We will discuss this and similar implications toward the end of the paper.

5.3.4 Summary

Through a crowd-sourcing methodology we have shown that there are substantial differences in how searchers refine their queries in different stages in a struggling task. These differences have strong connections with task outcomes, and there are particular pivotal queries that play an important role in task completion.

5.4 Predict Reformulation Strategy

Given the differences in query reformulation strategies and to help operationalize successful reformulations in practice, we develop classifiers to (1) predict inter-query transitions during struggling searches according to our intent-based schema (Table 5.2), and (2) identify pivotal queries within search tasks. This facilitates the development of anticipatory support to help searchers complete tasks. For example, if we predict a searcher wants to add an action to the query, we can provide query suggestions and auto-completions with actions. Our classifiers can also be applied retrospectively, e.g., to identify frequent transitions between queries and pivotal queries that form useful query suggestions or alterations.

Features. We include five sets of features, described in detail in Table 5.3. Some of the features relate to the characterizations that have been described in previous sections of the chapter. Query features are used to represent the query before a reformulation. Interaction features describe how a searcher interacted with the search engine result page (SERP). We saw in Figure 5.8 that the type of reformulation is dependent on the time elapsed since the previous query. After observing a new query, we can compute three new sets of features. First, the query transition features describe characteristics of the new query and low-level lexical measures of how the query has changed. Query similarity features describe the terms in a query have changed. For these features, similarity is measured using five different similarity functions, described in Table 5.4. Terms can match exactly, approximately, on their root form or semantically. Lastly, we include features that analyze the lexical reformulation of queries. For this, we used the procedure as described in Section 5.2.6 and a recent rule-based approach [98].

Experimental Setup. We frame this as a multi-class classification problem; one for each of the 11 query reformulation types. We use the 659 labeled tasks with a total of 1802 query transitions (Section 5.3). We perform ten-fold cross-validation over the tasks and use a RandomForest classifier, since it is robust, efficient and easily parallelizable. We experimented with other classifiers (including logistic regression and SVM), and none yielded better performance. We therefore only report the results of the RandomForest classifier. We evaluate our approach at different stages between two queries:

First Query: We only observe the first query and try to predict the next reformulation strategy. At this stage, search systems can tailor the query suggestions on the SERP.

First+Interaction: We observe the first query and interactions (clicks, dwell time). The searcher is about to type in a new query. At this stage, systems can tailor auto-completions for the next query.

Second Query: We observe both the first and second query and infer the reformulation strategy that the searcher applied. At this stage, search systems could re-rank results (or blend the results with those from other queries), and suggestions for the next query.

Concretely, the different stages mean that different groups of features become available (see Table 5.5 for the feature groups available at each stage). Finally, after the first query transition we add history features that represent the reformulation type and previous transition strategy. We report accuracy, area under the ROC curve (AUC) and F1 for our classifiers. F1 is computed as the harmonic mean of precision and recall per class. As a baseline we use the marginal (i.e., always predict the dominant class).

5.4.1 Prediction Results

Table 5.6 shows the results of our prediction experiments. The results are grouped by the transition and stage within a struggling search task. Our prediction experiments start with observing the first query (Q1) and end with observing the third query (Q3). We observe from the baseline results in Table 5.6 (lines 1 and 5) that this multi-class prediction task is a difficult problem. Apart from the first transition (where adding attributes is overrepresented, see Figure 5.13), the baseline approach of always predicting

Table 5.3: Features to predict reformulation strategy. The features marked with * are included for five similarity functions (listed in Table 5.5).

Name	**Description**
Query Features	
NumQueries	Number of queries in task
QueryCharLength	Query length in number of characters
QueryTermLength	Query length in number of terms
Interaction Features	
TimeElapsed	Time elapsed since query issued
ClickCount	Number of clicks observed since query
ClickDwelltime	Dwell time on clicked pages
QueryDwellTime	Dwell time on result page
Query Transition Features	
NewCharLength	New query length in number of characters
NewTermLength	New query length in number of terms
Levenshtein	Levenshtein edit distance in characters
normLevenshtein	Levenshtein edit distance as proportion of the longest query
commonCharLeft	Number of characters in common from left
commonCharRight	Number of characters in common from right
diffPOS:<type>	Difference in the number of terms that are identified in WordNet as belonging to a specific part of speech type: noun, verb, adjective, adverb
Query Similarity Features	
ExactMatch*	Number of terms that match exactly
AddTerms*	Number of terms not in previous query
DelTerms*	Number of terms not in new query
SubsTerms*	Number of terms substituted
QuerySim*	Proportion of terms exactly match
commonTermLeft*	Number of terms in common from left
commonTermRight*	Number of terms in common from right
Query Reformulation Features	
LexicalType	Lexical query reformulation type (see Table 5.1): new, back, morph, sub, general, specific
RuleBasedType	Lexical reformulation type using rule-based classifier of Huang and Efthimiadis [98]

Table 5.4: Similarity matching functions used to compare terms for query similarity features.

Name	Description
Exact	All characters match exactly
Approximate	Levenshtein edit distance is less than two
Lemma	Terms match on their lexical root or lemma form
Semantic	WordNet Wu and Palmer measure greater than 0.5 (measures relatedness using the depth of two synsets and the least common subsumer in Wordnet)
Any	Any of the above functions match

Table 5.5: Groups of features available at different stages.

Feature group	First Query	First+Interaction	Second Query
Query	✓	✓	✓
Interaction		✓	✓
Transition			✓
Similarity			✓
Reformulation			✓

the dominant class is only about 25% correct, and obviously not very informative to support a struggling searcher.

For the first query reformulation (Q1 to Q2, lines 1–4 in Table 5.6), our classifiers already improve the reformulation strategy prediction before the second query. While just observing the first query does not provide significant improvements on all metrics (line 2), observing clicks and dwell times increases the F1 score significantly from 20% to 34% (line 3). If a struggling searcher issues a second query, we can infer the applied strategy with a 43% accuracy and significantly better on all metrics (line 4). Turning to the second query reformulation (Q2 to Q3, lines 5–8 in Table 5.6), the baseline performance (line 5) is substantially lower as the dominant class is less prevalent. Directly after observing the second query and using the task history, we can predict the reformulation strategy with 46% accuracy (line 6). Observing the interactions with the second query does not substantially improve prediction performance (line 7). If we observe the third query, the reformulation strategy can be inferred with 55% accuracy (line 8). All classifiers for the second query transition (lines 6–8) perform substantially better than those for the first transition (lines 2–4). This suggests that understanding what strategy a searcher has used previously helps in predicting the next reformulation strategy that they will use.

Feature Analysis. We measure the feature importance as gini importance [30] averaged over all trees of the ensemble. Figure 5.15 visualizes the total importance of the six groups of features. For predicting the first transition (Q1 to Q2) directly after the first query, only the query features are available. The interaction features that become available for the first query are substantially more important. If we observe the second query, the similarity

Table 5.6: Results for predicting query reformulation strategy. Significant differences, tested using a two-tailed Fisher randomization test against row 1 for 2–4 and row 5 for 6–8, are indicated with △ ($p < 0.05$) and ▲ ($p < 0.01$).

Transition	Stage	Accuracy	F1	AUC
1. Q1 to Q2	Baseline	0.3736	0.2032	0.5000
2. Q1 to Q2	Q1	0.3679	0.3046▲	0.5322△
3. Q1 to Q2	Q1+Interaction	0.3698	0.3371▲	0.5589▲
4. Q1 to Q2	Q2	0.4302△	0.3916▲	0.6077▲
5. Q2 to Q3	Baseline	0.2095	0.0971	0.4908
6. Q2 to Q3	Q2	0.4644▲	0.4595▲	0.6799▲
7. Q2 to Q3	Q2+Interaction	0.4862▲	0.4826▲	0.6896▲
8. Q2 to Q3	Q3	0.5474▲	0.5321▲	0.7306▲

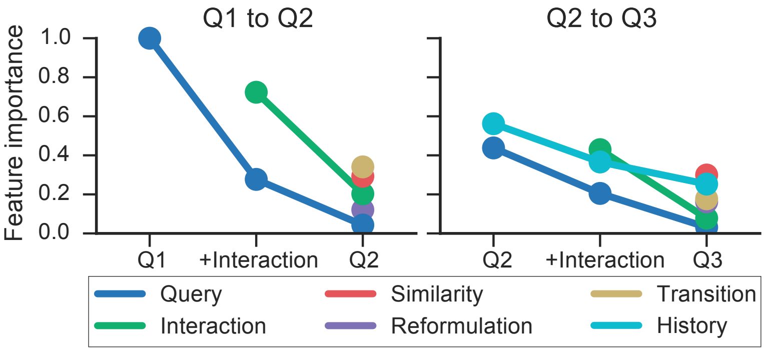


Figure 5.15: Feature group importance for Q1 to Q2 transition (left) and Q2 to Q3 transition (right).

and transition features are most important. The query features no longer contribute much to the classifier, as do the reformulation features. For the transition from the second to the third query (Q2 to Q3), the pattern is similar, but the history features contribute more.

Identifying the Pivotal Query. We are also interested in how accurately we can identify the pivotal query. We use a similar set-up as above, using subsets of the described features to identify the pivotal query in the 659 labeled tasks. Table 5.7 shows the results of this approach. The baseline always predicts the terminal query as the pivotal query (Figure 5.12). Using only the interaction and query features from Table 5.3 we significantly outperform this baseline in terms of F1 and AUC. Adding more features does not increase performance on any metric. Although our results are promising, they also suggest that identifying the pivotal query is difficult.

Table 5.7: Results for retrospectively identifying the pivotal query. Significant differences, tested using a two-tailed Fisher randomization test against row 1 are indicated with △ ($p < 0.05$) and ▲ ($p < 0.01$).

	Accuracy	**F1**	**AUC**
1. Baseline	0.6245	0.5091	0.7038
2. Query + Interaction	0.6352	0.5817▲	0.7296△

5.5 Discussion and Conclusions

Search engines aim to provide their users with the information that they seek with minimal effort. If a searcher is struggling to locate sought information, this can lead to inefficiencies and frustration. Better understanding these struggling sessions is important for designing search systems that help people find information more easily. Through log analysis on millions of search tasks, we have answered RQ3 and characterized aspects of how searchers struggle and (in some cases) ultimately succeed. We found that struggling searchers issue fewer queries in successful tasks than in unsuccessful ones. In addition, queries are shorter, fewer results are clicked and the query reformulations indicate that searchers have more trouble choosing the correct vocabulary.

We have shown significant behavioral differences given task success and failure. This informed the development of a crowd-sourced labeling methodology to better understand the nature of struggling searches. We proposed and applied that method to answer RQ4 and better understand the struggling process and where it became clear the search would succeed. This pivotal query is often the last query and not all strategies are as likely to be pivotal. We developed classifiers to accurately predict key aspects of inter-query transitions for struggling searches, with a view to helping searchers struggle less.

The research in this chapter has focussed on user behavior in a particular type of long search sessions in which we know that a user has trouble finding what they are looking for. Earlier in the thesis, in Chapter 3, we considered a different type of long sessions in which researchers where exploring large news collections. Among other things, our results in this chapter show that less successful struggling searchers use more spelling corrections, substitutions, completely new queries and returning to previous queries. It appears that less successful searchers experience more difficulty selecting the correct query vocabulary. Motivated by this, we continue the research in the thesis with a third and final research theme by considering a pro-active search setting, in which we will try to automatically generate queries and find relevant content for a user, based on the context of their search.

Our research has limitations that we should acknowledge. First, we focused on a very specific and very apparent type of struggling, indicated by limited activity for the first two queries within a task. More forms of struggling exist and they might exhibit different search behaviors. Our determinations of search success for the log analysis were based on inferences made regarding observed search activity, especially satisfied clicks based on dwell time. Although these have been used in previous work [72], and were validated by third-party judges as part of a dedicated judgment effort, there are still a number of factors that can influence landing page dwell time [114]. Finally, the crowd-sourced annotations were based on judgments from third-party judges and not the searchers themselves. While this methodology has been used successfully in previous work [92], methods to collect judgments in-situ can also be valuable [72].

Previous work has shown that it is possible to detect struggling automatically from behavior [92, 192]. Our focus has been on better understanding struggling during search and predicting query reformulation strategy. Ideally, a search engine would interpret the behavioral signals that indicate struggling and frustration to provide personalized help to searchers to help them attain task success. The types of support possible include:

Direct application of reformulation strategies. Demonstrating the capability to accurately predict the strategy associated with the next query reformulation (rather than syntactic transformations, as has traditionally been studied) allows us to provide situation-specific search support at a higher (more strategic) level than specific query formulations. For example, if we predict that a searcher is likely to perform an action such as adding an attribute, the system can focus on recommending queries with attributes in query suggestions or query auto-completions depending on when they are applied (and augmented with additional information about popularity and/or success if available from historic data). Sets of (query → pivotal query) pairs can also be mined from log data. Such queries may also be leveraged internally within search engines (e.g., in blending scenarios, where the results from multiple queries are combined [178]) to help generate better quality search result lists, or present them as suggestions to searchers.

Hints and tips on reformulation strategies. As we demonstrated, struggling searchers, especially those destined to be unsuccessful, are highly likely to re-query. Learning the relationship between task success and the nature of the anticipated query reformulation allows search systems to generate human-readable hints about which types of reformulations to leverage (e.g., "add an action" leads to the highest proportion of pivotal queries, per Figure 5.14), and propose them in real time as people are searching. Mining these reformulations retrospectively from log data also allows search systems to identify the most successful query transitions in the aggregate—rather than focusing on proxies for success, such as query or resource popularity [111, 226]. These reformulations can be learned from all searchers, or perhaps even more interestingly, from advanced searchers [2, 226] or domain experts [228].

Overall, accurate inferences and predictions about the nature of query reformulations can help searchers and search engines reduce struggling. Although our findings are promising, the nature and volume of our human-labeled data limited the types of the prediction tasks that we attempted. There are others, such as predicting search success given different query reformulation strategies that are interesting avenues for future work. Additional opportunities include working directly with searchers to better understand struggling in-situ, improving our classifiers, and experimenting with the integration of struggling support in search systems.

6
Real-Time Entity Linking based on Subtitles

In the third and final research theme in this thesis, we consider a pro-active search scenario, specifically in a live television setting, where we propose algorithms that leverage contextual information to retrieve diverse related content for a leaned-back TV viewer. Our work in this research theme is indirectly motivated by the observation in the previous chapter that less successful web searchers experience more difficulty in selecting the correct query vocabulary. In a pro-active search scenario we try to automatically generate queries to find relevant content for a user, based on the context of their search. We consider a live TV setting, where the main challenge is to find content while a story is developing. For live TV, subtitles are typically available (provided for the hearing impaired). Using this textual stream of subtitles, we can automatically (1) provide links to background information and (2) generate search queries to find related content. Both tasks have unique demands that require approaches that need to (1) be high-precision oriented, (2) perform in real time, (3) work in a streaming setting, and (4) typically, with a very limited context. By leveraging the textual stream of subtitles, we cast these tasks as information retrieval (IR) problems in a streaming setting. In this research theme, we consider both of these tasks in two research chapters. First, in this chapter, we consider the task of providing links to background information for live television.

Television broadcasts are increasingly consumed on an interactive device or with such a device in the vicinity. Around 70% of tablet and smartphone owners use their interactive devices while watching television [155]. These developments allow the television audience to interact with the content they are consuming, extending the viewing experience both live and on-demand. The interaction includes not only producing and consuming broadcast-specific social media, but it also caters for providing content that is created exclusively for the interactive device, such as additional background information. When an interactive device is used in this fashion, it is commonly referred to as a *second screen*.

For live television, edited broadcast-specific content that is meant be used on second screens is hard to prepare in advance and producing it can be very time-consuming. We present an approach for automatically generating links to background information in real time, to be used on second screens. Our approach automatically generates links to Wikipedia. This process is commonly known as *entity linking* and has received much attention in recent years [67, 139, 140, 148, 149]. Such links are typically explanatory, enriching the link source with definitions or background information [33, 94]. Recent work has considered entity linking for short user-generated texts such as queries and microblogs [26, 87, 139–141]. The performance of generic methods for entity linking deteriorates in such settings, as language usage is creative and context virtually absent.

We base our entity linking approach for television broadcasts on subtitles, thereby casting the task as one of identifying and generating links for elements in the stream of subtitles. Note that the subtitles are not actually shown, but only used as a textual stream to generate links that may then be shown through a visual representation, on a second screen or in an interactive video player, as sketched in Figure 6.1. Traditional document-based approaches to entity linking are not suited for this task, as links need to be generated continuously from this stream. On the other hand, using entity linking approaches for short text, that completely ignore the streaming nature of the material, would mean missing out on important contextual signals. Hence, in order for our entity linking approach to be effective in the context of second screens, it needs to be fast, able to disambiguate between candidate links in real time, and leverage streaming data so as to capture context. Furthermore, since viewers are dividing their attention between the actual broadcast and the second screen, the information that is being offered needs to be of high quality, i.e., have high precision.

We propose an entity linking approach for generating links from streaming text, consisting of two steps: (1) link retrieval and (2) link reranking. We use learning to rerank to improve upon a strong link retrieval approach. In addition, we model context explicitly in the reranking approach using a graph-based and a retrieval-based approach. These approaches are particularly appropriate in our setting as they allow us to combine a number of context-based signals in streaming text and capture the core topics relevant for a broadcast, while allowing real-time updates to reflect the progression of topics in the broadcast. Both context models are highly accurate, fast, and allow us to disambiguate between candidate links and capture the context as it is being built up.

We address the following research questions:

RQ5 Can we effectively and efficiently provide background information for a live television broadcast in real time using an entity linking approach and does explicitly modeling streaming context improve the effectiveness?

- **RQ5.1** What is the performance, in terms of effectiveness and efficiency, of a state-of-the-art retrieval model on the task of entity linking of streaming text?
- **RQ5.2** Is a learning to rerank approach with task-specific features able to improve in terms of effectiveness over the retrieval baseline, and, if so, at what computational costs?
- **RQ5.3** Can we leverage the streaming nature of a textual source, by modeling context explicitly, to improve the effectiveness of entity linking?

Our main contribution is a set of effective feature-based methods for performing real-time entity linking. For the link retrieval step, we explore multiple link retrieval and link pruning approaches and thoroughly evaluate the effects on both efficiency and effectiveness. For the link reranking step, we show how a learning to rerank approach for entity linking

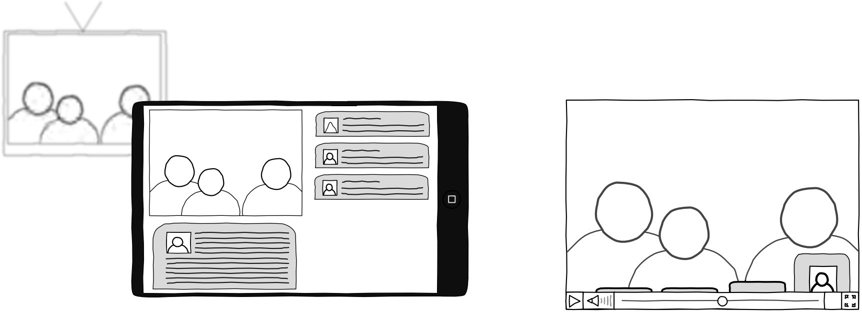


Figure 6.1: Sketches of a second screen (left) and an interactive video player (right) showing links to background information, synchronized with a television broadcast. Links pop up briefly when relevant and are available for bookmarking or exploring.

performs on the task of real-time entity linking, in terms of effectiveness and efficiency. We extend this approach with a graph-based and a retrieval-based method to keep track of context in a textual stream and show how this can further improve effectiveness. By investigating the effectiveness and efficiency of individual features we provide insight in how to improve effectiveness while maintaining efficiency for this task. Additional contributions include a formulation of a new task: entity linking of a textual stream, and the release of a dataset for this new task, including ground truth.

The remainder of this chapter is organized as follows, in Section 6.1 we formalize the task of entity linking of a textual stream; Section 6.2 describes our proposed method. In Section 6.3 we describe the experimental setup. We present our results in Section 6.4, after which we conclude. We have discussed related work on search in a streaming setting and link generation above in Section 2.4.

6.1 Task Definition

The task we consider is real-time entity linking of a textual stream. We link subtitles that come with television broadcasts—live or recorded—to Wikipedia articles. The identified links should be interesting and relevant for a wide audience. In the context of subtitles, we define a dynamic textual stream to be a source that continually produces "chunks" of text. Here, a *chunk* is the amount of subtitle text that can be displayed on a single screen (again, we are not assuming that the text is actually displayed, only that it comes with the broadcast). Chunks are relatively short and in our data contain approximately seven terms on average, as we will see below when we discuss our experimental setup. Therefore, chunks do not necessarily form a grammatical sentence. However, as these chunks are produced to be read in sequence, syntactic phrases generally do not cross chunk boundaries. Chunks form a growing sequence $S = \langle s_1, \ldots \rangle$ and our task is to decide, in real time, whether a link to Wikipedia should be created for chunk s_i and what the link target should be.

We generate a growing set of link candidates $C = \{c_1, \ldots, c_M\}$ that each link a specific anchor to a Wikipedia article $w \in W$. A *link candidate* $c_i \in C$ is a triple $((s_i, a), w)$ that links an "anchor" (s_i, a) to a target w, where an *anchor* is a term n-gram a within a chunk s_i. Each target w is a proper Wikipedia article, i.e., excluding redirect

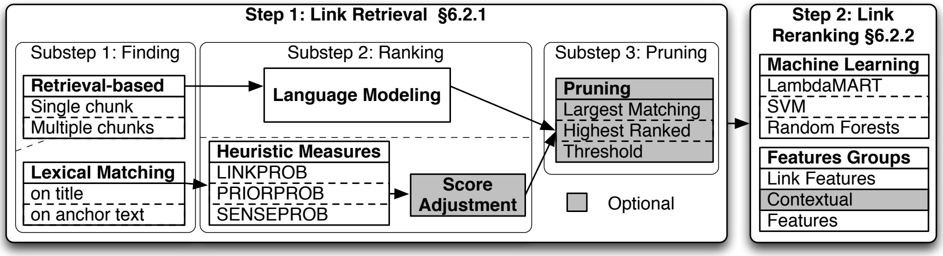


Figure 6.2: Graphical representation of our two-step real-time entity linking approach. A shaded background signifies an optional component and dashed lines indicate variants that are explained in detailed in the corresponding sections.

and disambiguation pages, taken from a set of Wikipedia articles W. A target is identified by its unique title on Wikipedia. In the dataset that we use in this chapter (see below) we have manually identified video segments of a television program; here, a video segment is a set of chunks that share a single topic. An interview with a guest in a talk show is a typical example of a video segment that we encountered.

6.2 Real-Time Entity Linking

Next, we introduce our approach to real-time entity linking. It consists of two steps: (1) link retrieval (Section 6.2.1), and (2) link reranking (Section 6.2.2). The first step is recall-oriented and aimed at finding as many link candidates as possible. The second step improves over this link retrieval step using a learning to rerank approach using a set of lightweight features. On top of that, we describe an extension to the learning to rerank approach that explicitly models context. Figure 6.2 shows a helicopter view of our approach.

6.2.1 Link Retrieval

The aim of the first step is to produce link candidates for a given chunk. Concretely, we are given a chunk s_i and need to retrieve a set of link candidates C that each link to a Wikipedia article w. We decompose this into three substeps: (1) finding link candidates (Section 6.2.1), (2) ranking link candidates, with optional score adjustment (Section 6.2.1), and (3) link candidate pruning (Section 6.2.1, optional).

Finding link candidates

We evaluate four approaches to finding link candidates: two retrieval-based and two based on lexical matching. The ones based on retrieval and those based on lexical matching call for different methods for ranking link candidates (described below in Section 6.2.1).

Finding link candidates through retrieval. Our retrieval-based approaches to link candidate finding consider the chunks of subtitles as queries to search over an index of Wikipedia articles. We either use a single chunk or multiple chunks to generate a query.

Algorithm 1: Lexical matching of anchor text

```
Input: Chunk $s_i$ consisting of sequence of words
Output: Set of link candidates $C = \{c_1, \ldots, c_M\}$ that each link a specific anchor
        $(s_i, a)$ to a Wikipedia article $w \in W$
$C \longleftarrow \emptyset$;
for $n \leftarrow$ number of words in $s_i$ to 1 do
    foreach $n$-gram $a$ in $s_i$ do
        if $a$ is anchor text for $w \in W$ then
            $C \longleftarrow C \cup \{((s_i, a), w)\}$
```

In our first retrieval-based model that considers a single chunk, each term in s_i is matched against all Wikipedia articles $w \in W$. Articles are included in the set of link candidates C if any term matches. In our second retrieval-based model, we consider multiple chunks $\{s_i, \ldots, s_j\}$ and generate a query from selected terms from these chunks. Up to 10 terms are selected with the highest TF.IDF score. This score is also used to weight each term. We use blocks of eight chunks of subtitles, which makes this link retrieval approach comparable to the retrieval approach of Blanco et al. [25], who use a sliding window of 30 seconds to generate queries to retrieve related news. We have used a similar approach in Chapter 3 for computing textual similarity on the task of related article finding in historical archives. In Chapter 7, this approach will be one of the baselines for finding content related to live TV.

Finding link candidates through lexical matching. The first lexical matching model we employ is based on the titles of Wikipedia articles. If a title of an article occurs in the chunks of subtitles, we consider that article to be a link candidate. Our second lexical matching approach is based on how links between Wikipedia articles are created. In this model, each Wikipedia article is represented by the anchors that are used to link to it within Wikipedia. To this end, we perform lexical matching of each n-gram a of chunk s_i with the anchor texts found in Wikipedia. This link candidate finding approach is described in Algorithm 1.

Ranking Link Candidates

The second substep of link retrieval is to rank the link candidates in C. The approaches based on retrieval and those based on lexical matching call for different approaches to ranking link candidates.

Retrieval-based approaches. The link candidates produced using the retrieval-based approaches are subsequently scored based on the sum of a language modeling score for each term. We refer to this scoring function as $SCORE(s_i, w)$. In the multiple chunk variant (described in Section 6.2.1), this is a weighted sum based on the TF.IDF score, referred to as $SCORE(\{s_{i-8}, \ldots, s_i\}, w)$. Concretely, we use a language model with a Dirichlet prior with parameter $\mu = 2000$.

Lexical matching-based approaches. In the anchor-based lexical matching model we use statistics on how anchor text is used on Wikipedia to form a ranking function. We also use this for title-based lexical-matching model, considering only anchors that match the title of an article. From these statistics we compute three heuristic measures that estimate the likelihood of a link candidate article w being the target of a link for a lexically matched anchor a: $\mathit{LINKPROB}$, $\mathit{PRIORPROB}$ and $\mathit{SENSEPROB}$.

$\mathit{LINKPROB}$ is the proportion of documents in which the anchor text a appears as anchor text for any link on Wikipedia:

$$\mathit{LINKPROB}(a) = \frac{\sum_{w \in W} |L_{a,w}|}{\sum_{w' \in W} n(a, w')}, \tag{6.1}$$

where $L_{a,w}$ denotes the set of all documents with at least one link with anchor text a linking to article w and $n(a, w)$ is 1 if n-gram a occurs in w (either as anchor text or not), otherwise it is 0.

$\mathit{PRIORPROB}$ is the proportion of documents where Wikipedia article w is the target when the anchor text a appears on Wikipedia:

$$\mathit{PRIORPROB}(a, w) = \frac{|L_{a,w}|}{\sum_{w' \in W} |L_{a,w'}|}, \tag{6.2}$$

where $L_{a,w}$ denotes the same as in Equation 6.1. This measure is sometimes referred to as $\mathit{COMMONNESS}$ [141]. The intuition is that link candidates with anchors that always link to the same target are more likely to be a correct representation than those for which the anchor text is used more often to link to other targets.

If we combine the two measures $\mathit{COMMONNESS}$ and $\mathit{LINKPROB}$, we have an estimate for the probability that an n-gram a is used as an anchor linking to Wikipedia article w. We define this combined probability as:

$$\mathit{SENSEPROB}(a, w) = \frac{|L_{a,w}|}{\sum_{w' \in W} n(a, w')}, \tag{6.3}$$

where $L_{a,w}$ and $n(a, w)$ denote the same as in Equation 6.1. Intuitively, you can regard $\mathit{LINKPROB}$ as a measure of how likely a label is to be a link, $\mathit{PRIORPROB}$ as a measure of how unambiguous a label is and $\mathit{SENSEPROB}$ as a measure combining both characteristics.

Note that for all three measures we use statistics for documents, not for individual links. The motivation for this is that the Wikipedia style guide specifies that only the first occurrence of a particular link target should be linked and subsequent links should not. In preliminary experiments, we have evaluated using these measures based on individual links and found that these perform worse in terms of effectiveness. We therefore only report on the document-based measures.

Score adjustment. All three heuristic measures for ranking link candidates from the approach based on lexical matching ($\mathit{LINKPROB}$, $\mathit{PRIORPROB}$ and $\mathit{SENSEPROB}$) are computed as a proportion over documents and represent some estimate of the likelihood of a link candidate occurring in Wikipedia articles. When a link candidate occurs infrequently on Wikipedia, this proportion can be substantially overestimated. For example, an n-gram that occurs only twice and is used as anchor text only once would get the same $\mathit{LINKPROB}$ as one that is used as an anchor 50 times out of 100 occurrences. Instead, we would like to balance this estimated proportion for the number of observations that we base it on. We can see these heuristic measures for ranking as binomial proportions and thus as binomial distributions. The estimate for a binomial proportion has less variance when it is based on more observations, but could have the same observed proportion. To balance the estimated proportion for the number of observations, we propose to adjust the ranking scores by using a lower confidence bound estimate of these proportions. Concretely, we use the lower bound of a Wilson 95% confidence interval [229]:

$$\frac{\hat{p} + \frac{z^2}{2n} - z\sqrt{\frac{\hat{p}(1-\hat{p})+\frac{z^2}{4n}}{n}}}{1+\frac{z^2}{n}}, \tag{6.4}$$

where $\hat{p}$ is the uncorrected proportion, n is the number of observations and $z = 1.96$ for a two-sided 95% confidence interval. We evaluate the effect of using this lower bound instead of the original measures for ranking on the retrieval effectiveness of all three measures. Below, we refer to this as applying a lower confidence bound (LCB) on the heuristic ranking measures and evaluate the approaches based on lexical matching with and without this LCB estimate.

Pruning Link Candidates

Before applying learning to rerank, we consider pruning the set of link candidates, because the processing time for a chunk increases with each link candidate we need to consider for reranking. A commonly used approach to limit the number of link candidates is to start with the link candidates for the largest matching anchor and subsequently not consider any constituent n-grams [141]. We consider this approach and also a logical alternative where we start with the link candidates for the highest ranked anchor according to the $LINKPROB$ measure and ignore all constituent n-grams as anchors. Thirdly, we consider pruning the link candidates based on a threshold on the $SENSEPROB$ measure.

6.2.2 Learning to Rerank

In this section, we describe our approach for reranking link candidates. The first step of our approach to real-time entity linking generates link candidates and rank these candidates initially. The second step is aimed at improving precision using a learning to rerank approach, that was effective on similar tasks [120, 141, 149]. For link candidates many ranking criteria are in play, making learning to rerank particularly appropriate. This approach is closely related to the learning to rank approach discussed above in Section 2.1.2, but differs in that, here, we base the reranking on an initial ranking.

Learning algorithm. We evaluate three machine learning approaches: Random Forests, SVM and LambdaMART. We use a decision tree based approach as it has outperformed Naive Bayes and Support Vector Machines (SVM) on similar tasks [141, 149], choosing Random Forests [29] as it is robust, efficient and easily parallelizable. Random Forests is an ensemble classifier based on bagging of many decision trees. A learning algorithm is applied multiple times to a bootstrap sample of the instances and a random subset of features. The results of these classifiers are then averaged, making it less prune to overfitting. Here, we compare Random Forests against two other approaches: SVM and LambdaMART.

Features. We use two distinct groups of features: (1) link features and (2) contextual features (described below in Section 6.2.2). The link features are lightweight features that can be computed online. Among others, we use variants of features proposed by Meij et al. [141], that are suited for our television broadcast context. These 30 features are listed in Table 6.1. $WIKISTATS_n(w)$ is an indication of the popularity of a Wikipedia article and is defined as the absolute number of visitors for an article w in the n days before the television broadcast.[1] $WIKISTATSTREND_{n,m}(w)$ is the relative number of visitors in the last n days, compared to the number of visitors in the last m days. This feature is intended to provide information on peaks in the number of visitors. The intuition is that these features help to identify popular current topics in the television broadcast.

Modeling Context

We extend the learning to rerank approach with additional, contextual features. In this section, we describe how we model context explicitly to compute these contextual features. Link generation methods that rely on an entire document are not suited for use in the context of streaming text. Such methods typically rely on comparing all link candidates within a document. They are therefore computationally expensive, due to the many comparisons that have to be made (see also Section 2.4.2). Since we want to generate links for each chunk, we would need to do these comparisons for each chunk in the stream, not just once as in a document-based entity linking setting.

What we need, instead, is a method to model context that can be incrementally updated and by which we can easily compute features for link candidates. We consider two such models: one retrieval-based and one that models context as a graph.

Retrieval-based context model. In Section 6.2.1 we presented a retrieval model for link candidates that uses multiple chunks to generate a query to search over an index of Wikipedia articles. We use the scores as generated by this retrieval model as features that represent the textual similarity of a target Wikipedia article with the current context. Concretely, for the target article w of each link candidate, we compute the retrieval score, $SCORE(\{s_{i-8}, \ldots, s_i\}, w)$, using language modeling on the textual content of the last eight chunks $\{s_{i-8}, \ldots, s_i\}$, as described in Section 6.2.1. This retrieval score is computed for each link candidate considered in reranking and included as an additional feature for reranking.

Graph-based context model. Our second context model uses a graph to maintain and update an up-to-date model of the link candidates considered. We model the context of a textual stream as an undirected graph as follows. A *context graph* $G = (V, E)$ comprises a set V of vertices and a set E of edges; vertices are either a chunk s_i, a target w or an anchor (s_i, a). Edges link each chunk s_i to s_{i-1}. Furthermore, for each anchor (s_i, a), there is an edge from (s_i, a) to s_i and one from (s_i, a) to w. The graph reflects the content of the textual stream and encodes the structure by connecting chunks. This results in a smaller distance, i.e., fewer edges between nodes, for things that are mentioned together. Furthermore, nodes for Wikipedia articles that are mentioned more often, will have more anchors connecting to them, making these nodes more central and thereby more important in the graph. Figure 6.3 shows an illustration of a context graph with three chunks.

[1] See `http://dumps.wikimedia.org/other/pagecounts-raw/`

Table 6.1: Link features used for the learning to rerank approach, grouped by type.

Anchor features	
$\mathit{LEN}_f(a)$	Number of f in the n-gram a, where $f \in$ {terms, chars}.
$\mathit{IDF}_f(a)$	Inverse document frequency of a in representation f, where $f \in \{\text{title}, \text{anchor}, \text{content}\}$.
$\mathit{SNIL}(a)$	Number of article titles equal to a constituent n-gram of a.
$\mathit{SNCL}(a)$	Number of article titles containing constituent n-gram of a.
Target features	
$\mathit{LINKS}_f(w)$	Number of Wikipedia articles linking to or from w, where $f \in$ {in, out} respectively.
$\mathit{REDIRECT}(w)$	Number of redirect pages linking to w.
Anchor + Target features	
$\mathit{TF}_f(a, w) = \frac{n_f(a,w)}{\lvert f \rvert}$	Relative phrase frequency of a in representation f of w, normalized by length of f, where $f \in$ {title, first sentence, first paragraph}.
$\mathit{POS}_1(a, w) = \frac{pos_1(a)}{\lvert w \rvert}$	Position of the first occurrence of a in w, normalized by the length of w.
$\mathit{EDIT}(a, w)$	Levenshtein edit distance between anchor a and title of w.
$\mathit{NCT}(a, w)$	Does a contain the title of w?
$\mathit{TCN}(a, w)$	Does the title of w contain a?
$\mathit{TEN}(a, w)$	Does the title of w equal a?
Statistical features, computed both as ratio of documents and occurrences	
$\mathit{LINKPROB}(a)$	Probability that a is used as anchor text in Wikipedia.
$\mathit{PRIORPROB}(a, w)$	Probability of w as target of a link with anchor text a.
$\mathit{SENSEPROB}(a, w)$	Probability of w as target of when text a occurs.
Wikipedia visitor statistics features (Wikistats)	
$\mathit{WIKISTATS}_n(w)$	Number of times w was visited in last $n \in \{7, 28, 365\}$ days.
$\mathit{WIKISTATSTREND}_{n,m}(w)$	Number of times w was visited in the last n days divided by the number of times w visited in last m days, where the pair $(n, m) \in \{(1, 7), (7, 28), (28, 365)\}$.

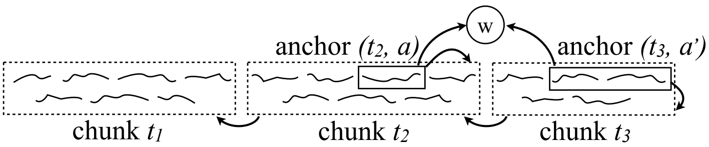


Figure 6.3: Illustration of a context graph with three chunks (s_1, s_2, s_3) and two link candidates with different anchors ((s_2, a) and (s_3, a')), but the same target w.

Algorithm 2: Updating the context graph.

```
Data: A context graph $G = (V, E)$, comprising a set $V$ of vertices or nodes and a
      set $E$ of edges. On initialization, $V = \emptyset, E = \emptyset$
Input: Set of link candidates $C = \{c_1, ..., c_M\}$ that each link a specific anchor
       $(s_i, a)$ to a Wikipedia article $w \in W$ for chunk $s_i$
Input: A ranking function $r$ assigning a score to a link candidate and a threshold $\tau$
Result: Nodes and edges added to $G$ for link candidates in $C$ if $r(a, w) > \tau$

$V \longleftarrow V \cup s_i$;
if $i > 0$ then
    $E \longleftarrow E \cup \{s_i, s_{i-1}\}$;
foreach $c = ((s_i, a), w) \in C$ do
    if $r(a, w) > \tau$ then
        $V \longleftarrow V \cup (s_i, a) \cup w$;
        $E \longleftarrow E \cup \{(s_i, a), s_i\} \cup \{(s_i, a), w\}$;
```

At the start of each video segment we initialize an empty context graph. The algorithm that describes how we update the context graph is described in Algorithm 2. We do not add vertices and edges to the context graph for every link candidate, for two reasons. First, the algorithm to find link candidates is recall-oriented and, therefore, produces many links for each chunk; a single usage of an anchor for a Wikipedia article is enough for it to be considered a link candidate. This introduces links that are very unlikely or trivial and can be considered noise. Second, given our objective to perform entity linking in real time, we need to limit the size of the context graph.

We select what link candidates to add to the context graph by placing a threshold, τ, on the probability of a link candidate being correct. For this we use the heuristic measures presented in Section 6.2.1: *LINKPROB*, *PRIORPROB* and *SENSEPROB*. We use *SENSEPROB* to select candidates to add to a context graph, as it captures both whether a link is correctly disambiguated and the likelihood of the anchor text being a link. We set the threshold value so that we can easily update the graph and compute the features from Table 6.2 during the time a particular subtitle is shown on the screen (roughly four seconds). On a small development set, this resulted in $\tau = 0.1$. We prune the graph for nodes that were added more than 100 chunks ago (~7 minutes on our dataset, see below). We revisit the choice of heuristic measure and the threshold value in Section 6.4.3.

To feed our learning to rerank approach with information from the context graph we compute a number of features for each link candidate. These features are described in

Table 6.2: Contextual features used for learning to rerank on top of the features listed above in Table 6.1.

Retrieval-based contextual features	
$SCORE(\{s_{i-8}, \dots, s_i\}, w)$	Retrieval score for article w based on a query generated from the last eight chunks $\{s_{i-8}, \dots, s_i\}$ using language modeling (see Section 6.2.1).
Graph-based contextual features	
$DEGREE(w, G)$	Number of edges connected to the node representing Wikipedia article w in context graph G.
$DEGREE-$ $CENTRALITY(w, G)$	Centrality of Wikipedia article w in context graph G, computed as the proportion of edges connected to the node representing w in G.
$PAGERANK(w, G)$	Importance of the node representing w in context graph G, measured using PageRank.

Table 6.2. First, we compute the degree of the target Wikipedia article in this graph. To measure how closely connected a target is, we compute degree centrality. Finally, we measure the importance of a target by computing its PageRank [169].

6.3 Experimental Setup

We describe the dataset used, our ground truth and metrics.

Dataset. To measure the effectiveness and efficiency of our proposed approach to entity linking, we use the subtitles of six episodes of a live daily talk show on Dutch television, called *De Wereld Draait Door*. This talk show covers current affairs in items of five to ten minutes, with various guests such as politicians, writers, etc. The episodes were broadcast between May 2010 and January 2011. The subtitles are generated during live broadcast by a professional and are intended for the hearing impaired. From these subtitles, video segments are identified, each covering a single item of the talk show. These video segments are based on the structure of the talk show. Video segments cover a single topic; their boundaries are manually identified during annotation.[2] Our dataset consists of 5,173 chunks in 50 video segments, with 6.97 terms per chunk. The broadcast time of all video segments combined is 6 hours, 3 minutes and 41 seconds.

Establishing ground truth. In order to train the supervised machine learning methods in our reranking approach and also evaluate the end result, we need to establish a gold standard. To this end, we have asked a trained human annotator to manually identify links that are relevant for a wide audience.[3] The subtitles are in Dutch, so we link to a Dutch

[2] We leave automatically identifying video segment boundaries in our dataset for future work and note that effective approaches exists [25, 95].

[3] To validate these manual annotations, we have asked a second annotator to annotate six video segments; 95.9% of links identified by the main annotator were also found by the second one.

version of Wikipedia.[4] Each video segment is assessed in sequence and links are identified by selecting anchors and a target Wikipedia article. If no target can be identified a link with a NIL target is created. A total of 1,596 links have been identified, 150 with a NIL target and 1,446 with a target Wikipedia article, linking to 897 unique articles, around 17.94 unique articles per video segment and 2.47 unique articles per minute.

Evaluation metrics. For the evaluation of our approaches, we are specifically interested in the ranking that is assigned to each link. We therefore regard the ranked list of all target Wikipedia articles for the link candidates that are produced by the baseline retrieval model for a video segment. Our learning to rerank approach assigns new ranks for the ranked list, but does not update the elements making up the list. We report average R-precision and mean average precision (MAP).

We also measure efficiency. We report the average classification time per chunk on a single core of an eight core machine. This classification time per chunk indicates how long it takes to produce links for one line of subtitles, after they appear on the screen. It should be noted, that *all* link features can be computed off-line, only requiring a simple lookup at runtime.

Features and learning methods. To compute our features, we use a Wikipedia dump from November 2011 ($\sim$1M articles) for which we calculate link statistics. We normalize all anchors and n-grams by removing all diacritical and hyphenation markers. For the $WIKISTATS$ features, we collect visitor statistics for Wikipedia on a daily basis and aggregate these per day, week, month, and year. This preprocessing makes all features fairly easy to compute at runtime.

The reranking step is evaluated using five-fold cross-validation at the video segment level, meaning that we test on five sets of ten video segments, training on the other 40. The Random Forests algorithm has two free parameters; the number of trees is set to 1500, based on preliminary experiments reported in Section 6.4.2, and we set K, the number of features to consider when taking a bootstrap sample, according to the common rule-of-thumb, to roughly 10% of the number of features [151].

Baselines. The task we proposed in Section 6.1 is to identify interesting links from a stream of subtitles that are relevant for a wide audience. For this task, no dedicated approaches exist. Two distinct approaches to address this using existing methods are to use entity linking or to use a retrieval approach that is suited for a streaming setting. Our experiments include a commonly used entity linking baseline [26, 87, 141] of a retrieval model using lexical matching on anchor text with $PRIORPROB$ as a ranking function, without score adjustment. This is typically referred to as the $COMMONNESS$ or $CMNS$ approach (see Section 6.2.1). As described in Section 6.2.1, our link retrieval approach that uses multiple chunks for retrieving link candidates is comparable to the retrieval approach of Blanco et al. [25], who use a sliding window of 30 seconds to generate queries to retrieve related news. We use this same approach as a baseline in the next chapter and use a similar retrieval approach for measuring textual similarity on the task of finding

[4] There is nothing in our approach, however, that is language specific. One could even argue that entity linking for Dutch is more difficult than for English as the Dutch version of Wikipedia contains fewer Wikipedia articles.

Table 6.3: Design of the experiments, indicating their relation to the research questions and the variants considered at each (sub)step. * *Machine learning approaches*.

	Research	Link Retrieval			Link Reranking	
Experiment	Question	Finding	Ranking	Pruning	ML*	Features
1. Candidate finding	RQ5.1	All	-	-	-	-
2. Pruning candidates	RQ5.1	All	-	All	-	-
3. Ranking candidates	RQ5.1	All	All	All	-	-
4. Link reranking	RQ5.2	Best	Features	Best	All	Link
5. Coordinate ascent	RQ5.2	Best	Features	-	Best	Link
6. Modeling context	RQ5.3	Best	Features	-	Best	All
7. Varying thresholds	RQ5.3	Best	Features	-	Best	All

related news articles in Chapter 3. Although here our setting is different, we consider this to be a representative baseline for an existing retrieval-based approach.

Experiments. Figure 6.2 showed the many variants we consider at each (sub)step of our approach. In our experimental design, we gradually build the full approach by carefully evaluating each step. Table 6.3 lists the experiments and what choices are made in terms of variants for each (sub)step. Next, we discuss each research questions and the corresponding experiments in more detail.

RQ5.1 What is the performance, in terms of effectiveness and efficiency, of a state-of-the-art retrieval model on the task of entity linking of streaming text?

To answer RQ5.1, we perform three experiments that only consider the link retrieval step. Experiment 1 is recall-oriented, evaluating variants of the finding link candidates substep on both efficiency and effectiveness (in terms of recall only). Experiment 2 investigates the influence of pruning on the number of link candidates to consider for reranking and the effect on recall. Experiment 3 is focused on the effectiveness of the ranking substep, including the optional score adjustment. Although the ranking substep is executed before the pruning substep, pruning has implications for both recall and for ranking effectiveness, we therefore consider pruning before ranking.

RQ5.2 Is a learning to rerank approach with task-specific features able to improve in terms of effectiveness over the retrieval baseline, and, if so, at what computational costs?

To answer RQ5.2, we perform two experiments that build on the first three experiments. In experiment 4, we contrast the learning to rerank approach against the best approach for link retrieval and compare three pruning methods and three machine learning models. The ranking functions of the first step are included as features in this second step. For analysis, in experiment 5 we take the best performing machine learning model and add groups of features in a coordinate ascent [145], i.e., adding the best feature group at each step.

RQ5.3 Can we leverage the streaming nature of a textual source, by modeling context explicitly, to improve the effectiveness of entity linking?

To answer RQ5.3, we continue with the best performing approaches for link retrieval and link reranking in two experiments. In experiment 6, we add additional features that explicitly model context and evaluate whether this improves effectiveness and what effect this has on efficiency. In experiment 7, we analyze the influence of choosing what to add to the context graph on effectiveness.

6.4 Results and Discussion

We discuss the outcomes of our experiments aimed at answering our research questions. First, we consider the performance of the link retrieval step (RQ5.1, §6.4.1), then link reranking step (RQ5.2, §6.4.2) and finally our explicit modeling of context (RQ5.3, §6.4.3).

6.4.1 Link Retrieval

Our first research questions considers the performance, in terms of effectiveness and efficiency, of the first step of our real-time entity linking approach on the task of entity linking of streaming text. The substeps of our link retrieval step are to first find link candidates, then rank these link candidates and finally prune the link candidates. We evaluate each substep separately in three experiments, starting with the recall-oriented first substep.

Experiment 1: link candidate finding. Lines 1–4 of Table 6.4 show the results of experiment 1. Lines 1 and 2 show the retrieval-based models. We observe a recall of 56% for the retrieval approach that uses a single chunk, with an average retrieval time per chunk of 121 milliseconds. Using multiple chunks is substantially faster (because it needs fewer retrieval queries than one for each single chunk), but obtains a mere 36% recall.

The retrieval models based on lexical matching are listed in lines 3 and 4. Lexical matching of Wikipedia article titles (line 3) produces a limited number of link candidates (2,608) and fails to recall many of the relevant targets (only 40%). The lexical matching approach that uses anchor text (line 4) produces 59,410 link candidates, including 732 known targets that are in the ground truth (an average recall of 0.8493). Analysis shows that without normalization of anchor text that an 18 relevant targets are not recalled while producing 3,580 fewer link candidates (not listed in Table 6.4). We note that recall is comparable to the numbers reported in the literature for approaches that use lexical matching on anchor text for entity linking in documents [67, 149].

Experiment 2: pruning link candidates. With nearly 60,000 link candidates, there is a clear need for ranking, but also for pruning the link candidates. As pruning has implications for both recall and for ranking effectiveness, we discuss the results on the pruning substep (experiment 2) before the experimental results on the ranking substep. Lines 5–7 of Table 6.4 show the impact on recall for the three proposed pruning approaches. We observe that not considering link candidates for constituent n-grams substantially shrinks the set of link candidates by 84% (lines 5 and 6). The commonly used approach of first considering the largest matching n-gram (line 5) has higher recall than considering the highest ranked anchor first (line 6). The best tradeoff in terms of recall versus candidates

Table 6.4: Effectiveness and efficiency of the variants of the recall-oriented link candidate finding substep. **Include only candidates with a score adjusted SENSEPROB of at least 0.002.*

	Average processing time	Set-based	Relevant	
	per chunk (in ms)	Recall	Targets	Candidates
1. Retrieval per single chunk	121	0.5576	480	39,239
2. Retrieval using multiple chunks	32	0.3595	316	13,763
3. Lexical match on title	34	0.3997	342	2,608
4. Lexical match on normalized anchor	40	0.8493	732	59,410
Pruning link candidates				
5. Largest matching n-gram only	41	0.7165	631	9,669
6. Highest ranked n-gram only	41	0.7165	625	9,610
7. Threshold on $SENSEPROB^*$	42	0.8016	693	9,378

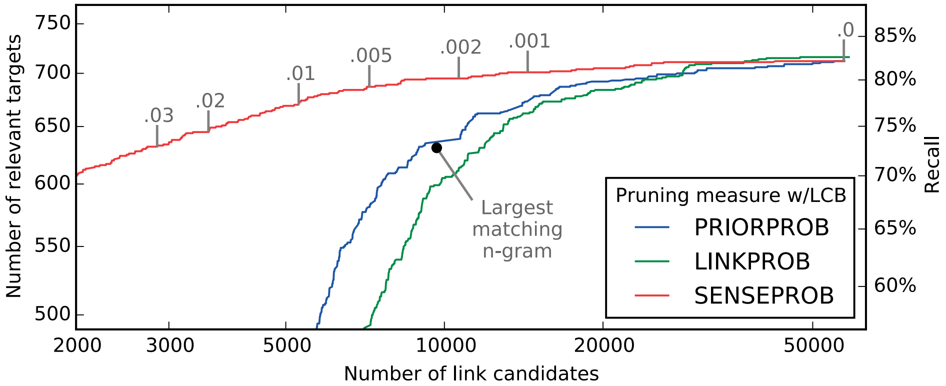


Figure 6.4: The influence of pruning with a heuristic measure threshold on recall and candidates.

is achieved when using a threshold on the $SENSEPROB$ measure. We observe on line 7 that we can maintain a recall of 80% while reducing the number of link candidates considered by 84%. We further investigate this tradeoff based on a score adjusted measure threshold in Figure 6.4. For nearly every reduction in the number of link candidates, the $SENSEPROB$ measure reduces the number of link candidates with higher recall than the other two measures. When presenting the results for subsequent (sub)steps of our approaches, we will continue to analyze the influence of pruning by largest matching n-gram and with a threshold, both on effectiveness and efficiency.

Experiment 3: ranking link candidates. Table 6.5 shows the results of experiment 3, concerning the initial ranking of our retrieval models. The two retrieval-based ranking approaches (lines 1 and 2) do not provide rankings that are up to par. We observe lower effectiveness as measured by R-precision and MAP compared to nearly all approaches

Table 6.5: Effectiveness of ranking link candidates for link retrieval using either retrieval (lines 1–2) or lexical matching on anchor text (lines 3–11) for link candidate finding. For the latter, three ranking approaches, each with and without score adjustment, as well as three pruning approaches are listed (no pruning, keeping only the largest matching n-gram or only candidates with a score adjusted $SENSEPROB$ of at least 0.002). The best result per column is highlighted in **bold**.

	Retrieval approach		R-Prec	MAP		
1.	Single chunk		0.0798	0.0515		
2.	Multiple chunks		0.1088	0.0732		
	Lexical matching on anchor		without LCB		with LCB score adjustment	
	Ranking	Pruning	R-Prec	MAP	R-Prec	MAP
3.	$PRIOR$	-	0.0664	0.0824	0.2496 (+276%)	0.2122 (+158%)
4.	$LINK$	-	0.3391	0.2067	0.3026 (−11%)	0.2817 (+36%)
5.	$SENSE$	-	0.4988	0.4561	0.5479 (+10%)	**0.5462** (+20%)
6.	$PRIOR$	threshold	0.1546	0.1533	0.2558 (+65%)	0.2287 (+49%)
7.	$LINK$	threshold	0.4185	0.3146	0.4404 (+5%)	0.4492 (+43%)
8.	$SENSE$	threshold	0.4988	0.4545	0.5479 (+10%)	0.5446 (+20%)
9.	$PRIOR$	largest	0.0690	0.0771	0.2707 (+292%)	0.2259 (+193%)
10.	$LINK$	largest	0.5262	**0.4601**	**0.5517** (+5%)	0.5159 (+12%)
11.	$SENSE$	largest	**0.5326**	0.4600	0.5439 (+2%)	0.5164 (+12%)

based on lexical matching of anchor text. Using multiple chunks for retrieval is more effective than using only a single chunk.

The anchor-based retrieval models (listed in lines 3–11) achieve reasonable effectiveness scores. Our ranking approaches try to balance the likelihood of a link being correct with a measure for link-worthiness. The $SENSEPROB$ measure does this by combining $PRIORPROB$ and $LINKPROB$ directly. We observe in lines 3–5 that without any pruning, this combined approach is most effective. Applying a threshold on $SENSEPROB$ substantially improves the effectiveness of the other two ranking approaches (lines 6 and 7), but not beyond using their combination (line 8).

Applying the largest matching n-gram pruning approach has two effects, it prunes any constituent n-gram anchors and it keeps only the most common target article as a link candidate. The combination of this pruning approach and $LINKPROB$ ranking is very comparable to ranking on $SENSEPROB$ and we see similar performance for both in lines 10 and 11. Line 9 shows the effectiveness of using $PRIORPROB$ heuristic measure for ranking and pruning the candidates by taking only the largest matching n-gram. This is the most commonly used heuristic-based approach, $CMNS$, and our entity linking baseline. The effectiveness of this ranking approach in our setting is lower than the numbers reported on web search queries [26, 87] and on microblog data [141]. Compared to these short text settings, it is more important in our setting to decide on whether or not a particular anchor is link-worthy. This is not captured in the baseline approach and clearly in our setting we need the information from the $LINKPROB$ measure.

The last two columns of Table 6.5 show the effectiveness of the ranking heuristic when

Table 6.6: Effectiveness of learning to rerank for link candidates for different machine learning and pruning approaches. Significant differences, tested using a two-tailed paired t-test, are indicated for lines 2–10 tested against line 1 at $p < 0.05$ ($^{\triangle}$) or $p < 0.01$ ($^{\blacktriangle}$).

Pruning	Approach	Average processing time per chunk	R-Prec	MAP
1. All	*SENSEPROB* w/LCB	40 ms	0.5479	0.5462
2. Largest	LambdaMART	79 ms	0.5460	0.5342
3. Threshold	LambdaMART	82 ms	0.5823$^{\triangle}$	0.5792
4. All	LambdaMART	124 ms	0.5870$^{\blacktriangle}$	0.5686
5. Largest	Support Vector Machines	54 ms	0.5789$^{\blacktriangle}$	0.5524
6. Threshold	Support Vector Machines	57 ms	0.5934$^{\blacktriangle}$	0.5924$^{\blacktriangle}$
7. All	Support Vector Machines	99 ms	0.5808$^{\blacktriangle}$	0.5754$^{\blacktriangle}$
8. Largest	Random Forests	58 ms	0.6089$^{\blacktriangle}$	0.5924$^{\blacktriangle}$
9. Threshold	Random Forests	61 ms	0.6272$^{\blacktriangle}$	0.6202$^{\blacktriangle}$
10. All	Random Forests	103 ms	0.6357$^{\blacktriangle}$	0.6213$^{\blacktriangle}$

we use a lower confidence bound (LCB) estimate on the proportions from Equation 6.4. We observe substantial improvements for nearly all combinations. *PRIORPROB* seems to benefit most from applying the LCB estimate. We observe the highest MAP and second best R-precision for rankings based on *SENSEPROB* (line 5). Applying the threshold-based pruning approach for this ranking function, simply comes down to applying a cutoff for the ranking and we observe no substantial impact on the ranking effectiveness if this threshold is applied (line 8).

Summary. As to our first research question, the state of the art retrieval models that we use all perform well on the task of entity linking streaming text in terms of efficiency. The performance in terms of effectiveness of lexical matching on anchor text is strong, with high recall-scores. In terms of precision these numbers are comparable to the literature, while leaving room for improvement.

6.4.2 Link Reranking

Next we consider the performance of the link reranking step (Section 6.2.2).

Experiment 4: link reranking. We contrast the performance in terms of effectiveness and efficiency before and after reranking. We compare to the best performing ranking approach from the previous experiment, using lexical matching based on anchor text with ranking on *SENSEPROB* with score adjustment and no pruning. We point out that this approach already obtained more than double the effectiveness scores compared to both baselines. Table 6.6 shows the effectiveness and efficiency of the learning to rerank approaches compared to the initial retrieval model (line 1). A model using Random Forests clearly outperforms LambdaMART and Support Vector Machines regardless of the pruning method used. Considering pruning, we observe substantially lower runtime per

Table 6.7: Entity linking results ranked by classification time. Significant differences, tested using a two-tailed paired t-test, are indicated for lines 2–6 tested against line 1 at $p < 0.05$ ($^{\triangle}$) or $p < 0.01$ ($^{\blacktriangle}$).

Approach	Average processing time per chunk	R-Prec	MAP
1. *SENSEPROB* w/LCB	40 ms	0.5479	0.5462
Learning to rerank (L2R) using Random Forests (listing feature groups used)			
2. L2R with only Statistical features	93 ms	$0.5821^{\triangle}$	$0.5795^{\triangle}$
3. + Anchor features	94 ms	$0.6207^{\blacktriangle}$	$0.6199^{\blacktriangle}$
4. + Anchor+Target features	99 ms	$0.6282^{\blacktriangle}$	$0.6248^{\blacktriangle}$
5. + Target features	102 ms	$0.6288^{\blacktriangle}$	$0.6273^{\blacktriangle}$
6. + Wikistats features	103 ms	$0.6357^{\blacktriangle}$	$0.6213^{\blacktriangle}$

chunk for Random Forests with only a small drop in effectiveness for the threshold-based pruning. For LambdaMART and SVM, we even observe increased R-Precision and MAP scores when pruning. In terms of efficiency, SVM is clearly the fastest approach. The more effective Random Forests approach is only 5 ms slower, even without pruning. LambdaMART adds another 21 ms, without improvements in effectiveness. As substantially more computational time is spent on computing features (48 ms without pruning) than on applying the reranking model, Random Forest seems to be the best choice.

We see that learning to rerank significantly boosts effectiveness over the initial link retrieval step, while keeping the average classification time per chunk at around 100 milliseconds or even less when carefully pruning the link candidates. This shows that classification in a streaming text setting is possible in near real time. In the following experiments, we refer to line 10 as *the learning to rerank approach.*

Experiment 5: coordinate ascent. Lines 2–6 in Table 6.7 show the results of experiment 5, where we add groups of features in a coordinate ascent [145], i.e., adding the best feature group at each step. We observe that leaving out specific feature groups can save computational time at limited cost in terms of effectiveness. Line 6 in the table concerns the *WIKISTATS* features: while they may gain slightly in terms of R-Precision, this comes at a big increase in pre-processing effort, as we need to collect visitor statistics for Wikipedia on a daily basis.

As an aside, we analyze the influence of one parameter for the Random Forest algorithm, the number of trees. We evaluate the effectiveness by taking the average of five runs with all features in Table 6.1. Figure 6.5 shows the results. The effectiveness increases as the number of trees increases and reaches a plateau at a value of about 150, indicating that a value of 1500 is a very safe setting and a few milliseconds of processing time can be won by decreasing this.

Summary. Our second research question is whether a learning to rerank approach would be able to outperform our retrieval baseline on the task of entity linking of streaming text. The results for our learning to rerank approach show that it can be highly effective and

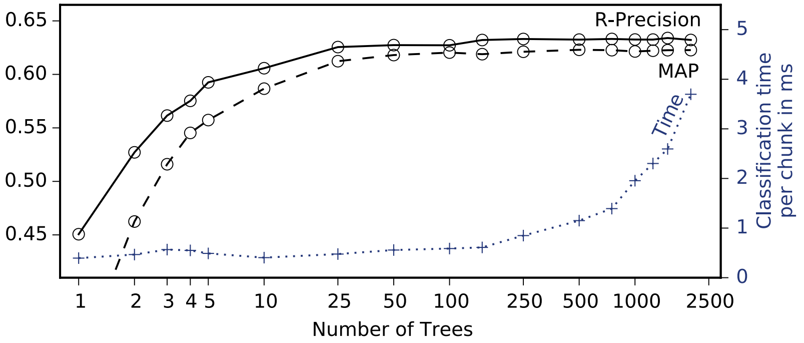


Figure 6.5: Analyzing the influence of the number of trees on efficiency and effectiveness as measured in terms of R-precision and MAP.

significantly improve over the retrieval approaches. We can achieve this high effectiveness at an average online classification time of around 100 milliseconds, making the learning to rerank approach efficient and suited for usage in real time.

6.4.3 Modeling Context

We turn to our third research question, and determine whether explicitly modeling context improves effectiveness on the entity linking task in experiment 6. We compute additional, contextual features based on the explicit model and include these in the link rerank step.

Experiment 6: modeling context. We first evaluate three features for the context graph (listed in Table 6.2). The results for the learning to rerank runs with these features added are listed in Table 6.8. Compared to the learning to rerank approach in line 2, we are able to achieve significantly higher performance in MAP, but not for R-Precision. All three context graph features (lines 3–5) improve effectiveness, with $DEGREE$ achieving the best score for R-Precision and MAP. The fact that we significantly improve on MAP, but not significantly on R-Precision indicates that mostly recall is improved by each of the three context graph features.

The combination of the three context features does not yield further improvements in effectiveness over the individual context graph features (as shown in line 6). To investigate why the combination does not improve effectiveness, we look at the result of a hypothetical oracle that picks the best of the result for each video segment. This gives us a ceiling value for effectiveness for the context graph setting that is very close to the value achieved by the individual feature runs, suggesting that the three features measure qualitatively similar things in the context graph.

Using the retrieval scores based on multiple chunks (line 8) improves effectiveness even more than graph-based context features. The combination of the two context approaches in line 9 achieves the best MAP scores. This suggests that both context models are complementary. To verify this, we analyze the difference in effectiveness per video segment for each of the two context models between the learning to rerank approach and its extensions with explicit contextual information. This difference is plotted in Figure 6.6.

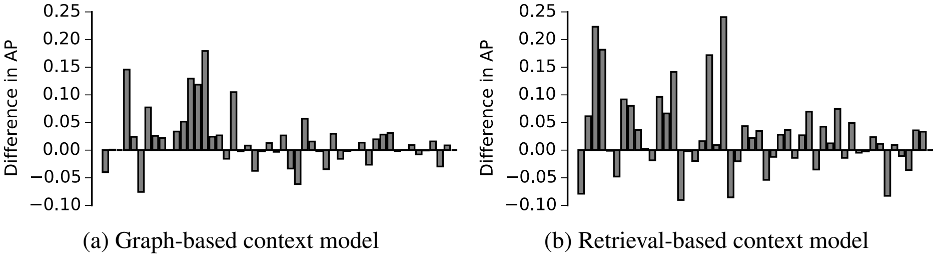
Table 6.8: Entity linking results for the graph-based context model, an oracle run (indicating a ceiling), the retrieval-based context model and a combined context model. Significant differences, tested using a two-tailed paired t-test, are indicated for lines 2–6 with $^{-}$ (none), $^{\vartriangle}$ ($p < 0.05$) and $^{\blacktriangle}$ ($p < 0.01$); the position of the symbol indicates whether the comparison is against line 1 (left most) or line 2 (right most).

	Average processing time per chunk	R-Prec	MAP
1. Heuristic ranking	40 ms	0.5479$^{-\blacktriangledown}$	0.5462$^{-\blacktriangledown}$
2. Learning to rerank (L2R)	103 ms	0.6357$^{\blacktriangle-}$	0.6213$^{\blacktriangle-}$
Learning to rerank + one context graph feature			
3. L2R + *DEGREECENTRALITY*	110 ms	0.6335$^{\blacktriangle-}$	0.6371$^{\blacktriangle\vartriangle}$
4. L2R + *DEGREE*	117 ms	0.6367$^{\blacktriangle-}$	0.6384$^{\blacktriangle\vartriangle}$
5. L2R + *PAGERANK*	131 ms	0.6367$^{\blacktriangle-}$	0.6349$^{\blacktriangle\vartriangle}$
Learning to rerank + three context graph features			
6. L2R+*DEGREE*+*PAGERANK* +*DEGREECENTRALITY*	133 ms	0.6276$^{\blacktriangle-}$	0.6388$^{\blacktriangle\vartriangle}$
7. Oracle picking the best out of lines 3–5 for each video segment		*0.6425*	*0.6469*
Learning to rerank + retrieval-based context features			
8. L2R + *SCORE* (multiple chunks)	137 ms	0.6404$^{\blacktriangle-}$	0.6482$^{\blacktriangle\vartriangle}$
9. L2R + all contextual features	167 ms	0.6390$^{\blacktriangle-}$	0.6548$^{\blacktriangle\blacktriangle}$

Comparing the individual improvements of both context models, we see substantial differences in performance per segment. Combining these features indeed leads to further improved performance.

To further investigate where the inclusion of contextual information helps and where it hurts, we analyze the difference in effectiveness per video segment between the learning to rerank approach and its extensions with the both context models (lines 2 and 9 of Table 6.8 respectively). Out of the 50 video segments, 41 show an increase and nine a decrease in effectiveness. We plot the difference in effectiveness against the number of target Wikipedia articles per video segment in Figure 6.7. We observe that for video segments with more links, the extended approach with our context models consistently improves effectiveness. This indicates that having more context to work with improves the effectiveness of the context models. On the other hand, we can also observe improvements for the video segments with fewer targets. This indicates that the context models are also able to improve effectiveness even when context is limited.

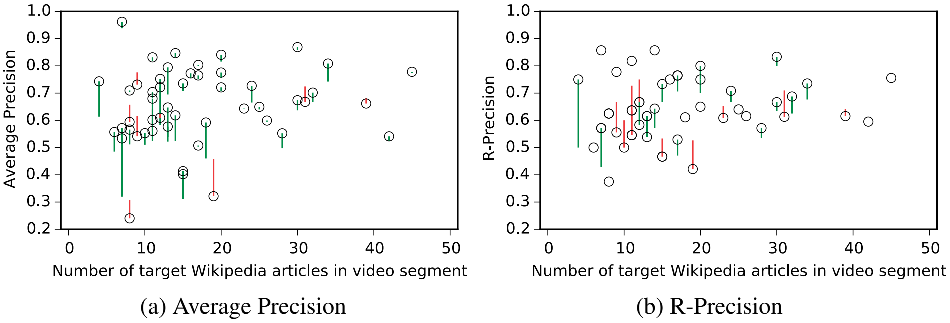
In Figure 6.7, two video segments stand out by showing relatively low effectiveness scores for both approaches (R-precision below 0.4). Neither video segment contains many links. Looking at their content, these are understandably hard cases as they discuss rare or even obscure news events (one covers the ups and downs of a homeless person, the other a less popular sports contest). Finally, most video segments that show a decrease



(a) Graph-based context model

(b) Retrieval-based context model

Figure 6.6: Analyzing the difference in effectiveness per video segment for each context approach, between the learning to rerank approach and the context approach. Effectiveness is measured by AP, for R-precision we observe qualitatively similar patterns. Bars are sorted from left to right in increasing performance for the learning to rerank approach.



(a) Average Precision

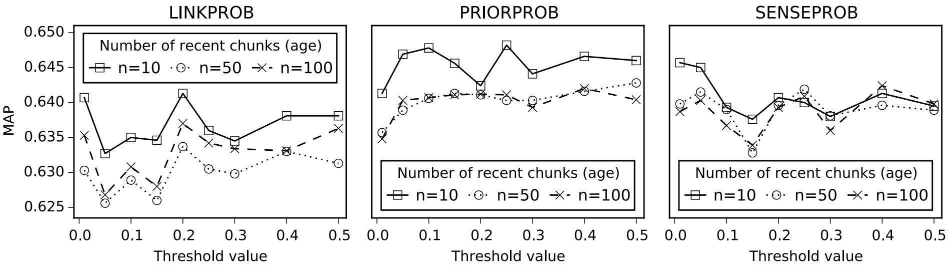
(b) R-Precision

Figure 6.7: Analyzing the difference in effectiveness in terms of AP (left) and R-Precision (right) per video segment, between the learning to rerank approach and the combined context model (using both textual and graph-based context features). For each video segment, a line starts at the effectiveness value for the learning to rerank approach, leading to a circle indicating the effectiveness for the combined context model approach.

in effectiveness under the graph-based context model already have a relatively high effectiveness in the learning to rerank approach.

In terms of efficiency, the $PAGERANK$ feature is computationally more expensive than the other two, but classification time is still low enough for this approach to be used in a real-time setting. Interestingly, the relatively simple to compute $DEGREE$ feature performs relatively well. Computing the retrieval-based context features takes an additional 34 ms, but we note that these features could be processed entirely in parallel to the lexical matching link generation approach, making the additional processing costs negligible.

Experiment 7: To add or not to add? The influence of the decision on what links to include in the context graph, as described in Section 6.2.2, deserves further attention in a final experiment (experiment 7). First, we use a threshold for what to add to the graph and second, we prune the graph based on age (defined as the n most recent chunks included in the context graph). We consider the effectiveness of three threshold



(a) *LINKPROB* (b) *PRIORPROB* (c) *SENSEPROB*

Figure 6.8: Analyzing the influence of a threshold on one of the three threshold functions for context graphs for the n most recent chunks, measured by MAP. For R-precision we observe no meaningful influence of the context features.

functions: *SENSEPROB*, *PRIORPROB*, *LINKPROB*, with threshold values $\tau \in [0, 0.5]$. Furthermore, we consider three values for the age n: 10, 50, and 100.

Figure 6.8 shows the results. For the *PRIORPROB* threshold function (Equation 6.2), increasing τ improves results until $\tau = 0.4$. Intuitively, this makes sense, as more ambiguous link candidates have lower *PRIORPROB* values. For the *LINKPROB* (Equation 6.1) and *SENSEPROB* (Equation 6.3) threshold functions, effectiveness is high for low values and there seems to be a sweet spot for τ between 0.1 and 0.3. The influence of the age value n seems to be less substantial, but there is indication that a lower value can give better performance. The *SENSEPROB* threshold function we proposed is most effective with a lower value for τ. The differences we observe in effectiveness for different threshold functions, values and ages are small (less than 0.03 difference in MAP between the best and worst scores), we therefore conclude that a low threshold value for either *SENSEPROB* or *PRIORPROB* with a lower number of recent chunks is a good choice.

Summary. Our third research question concerns whether we can improve the effectiveness of our learning to rerank approach if we model context explicitly, leveraging the streaming nature of a textual source. The results in Table 6.8 show that we can significantly improve the performance of our learning to rerank approach, while maintaining efficiency at an acceptable level, if we add features that were computed from the explicit context models.

6.5 Conclusion

Motivated by the rise in so-called second screen applications we introduced a new task: real-time entity linking of streaming text. We have created a dataset for this task[5] and have asked:

[5] The dataset (described in Section 6.3) is made available to the research community; it consists of more than 1,500 manually annotated links in over 5,000 subtitle chunks for 50 video segments. See: `http://ilps.science.uva.nl/resource/oair-2013-feeding-second-screen`

RQ5 Can we effectively and efficiently provide background information for a live television broadcast in real time using an entity linking approach and does explicitly modeling streaming context improve the effectiveness?

We have shown that learning to rerank can be applied to significantly improve an already competitive retrieval baseline and that this can be done in real time. Careful pruning and leaving out several features that are heavy to compute does not significantly decrease effectiveness, while substantially increasing efficiency.

Additionally, we have shown that by modeling context explicitly we can significantly improve the effectiveness of this learning to rerank approach. The proposed retrieval-based and a graph-based methods to keep track of context are especially well-suited for the streaming text, as we can incrementally update the context model. This makes these context models suitable for real-time applications. We have made our entire approach for real-time entity linking publicly available as a webservice and have released all code to run our approach as open-source.[6]

As to future work, we have shown that selecting which candidate links to add to the context graph is an important choice. An interesting follow-up to this observation is to further improve the representativeness of the context graph, for example by semantically enriching the graph, by weighting the edges of the graph, accounting for the quality of the evidence collected. We can further extend the enrichment to encode more information, e.g., by using the Wikipedia link structure. Adding edges in the context graph for links between Wikipedia pages could improve the coherence of the context graph. Future work could show whether the task of entity linking in streaming text benefits from such enrichments.

The dataset used in this study consists of subtitles from video segments to talk shows. We have chosen talk shows because they cover a range of topics, mostly current. However, our approach is not specific for talk shows and it will be interesting to evaluate this approach on different types of television broadcast, such as live events and sports. Furthermore, with advances in automatic speech recognition (ASR), combining our approach with the output of an ASR-system may provide an effective solution when manually created subtitles are not available.

In our experiments, a retrieval-based link generation approach proved less effective than one based on lexical matching of anchor text. The latter approach can benefit from information specific to the task of entity linking (i.e., the anchor text used on Wikipedia). In contrast, a retrieval-based approach is more general and therefore more suited for other, related tasks. In the next chapter, we revisit the retrieval-based approach for finding background information for the task of finding related video content.

[6] The approach described in Section 6.2 is implemented in a framework written in Python that provides a real-time entity linking API. For code, documentation and a webservice, see: `http://semanticize.uva.nl`. Appendix A details the framework and the usage of the webservice.

7

Dynamic Query Modeling for Related Content Finding

With this final research chapter of the thesis, we conclude the research within the third research theme, pro-active search for live TV. In a similar setting as in the previous chapter, we consider a new task: finding video content related to a live television broadcast for which we leverage the textual stream of subtitles associated with the broadcast. This task is more general than the task considered in Chapter 6: generating links to Wikipedia as background information. In this chapter, we turn to reinforcement learning to directly optimize the retrieval effectiveness of queries generated from the stream of subtitles. This approach can be applied to any target collection, not just to encyclopedic articles. As in the previous chapter, our approach needs to be highly efficient to be used in a live television setting, i.e., in real time. We ask the following research question:

RQ6 Can we effectively and efficiently find video content related to a live television broadcast in real time using a query modeling approach?

We model this task as a Markov decision process and propose a method that uses reinforcement learning to directly optimize the retrieval effectiveness of queries generated from the stream of subtitles.

As explained in Chapter 6, our research within this research theme is motivated by the changing way in which we watch television. A growing number of people do so either on an interactive device or with such a device nearby [156]. Razorfish [180] found that 38% of "mobile multitaskers" access content that is related to the TV program they are watching. The consumption of TV programs can trigger searches for related information or content, e.g., for a broader perspective or to dive deeper into a topic. Thus, we require effective methods to support these emerging information needs. Since people increasingly expect to have related content directly available instead of having to search for it—especially in a TV watching setting—we need to minimize the disruption of having to manually search for related content. An ideal system would therefore present related content in an automatic and timely fashion.

This new and emerging television landscape is beginning to generate new and interesting information retrieval problems. For instance, Blanco et al. [25] perform text-based retrieval in a TV setting, and in the previous chapter of this thesis, we considered the

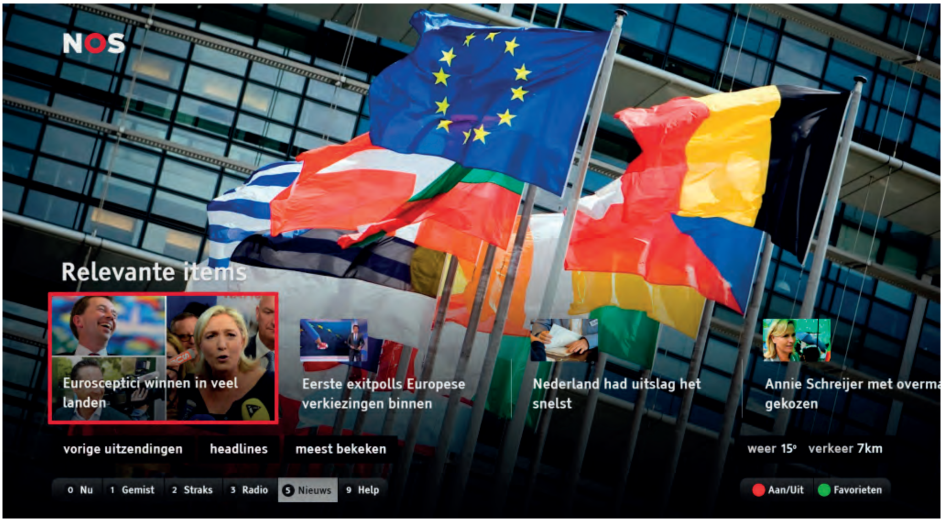

NOS
Relevante items
Eurosceptici winnen in veel landen
Eerste exitpolls Europese verkiezingen binnen
Nederland had uitslag het snelst
Annie Schreijer met overma gekozen
vorige uitzendingen
headlines
meest bekeken
weer 15°
verkeer 7km
0 Nu
1 Gemist
2 Straks
3 Radio
5 Nieuws
9 Help
● Aan/Uit
● Favorieten

Figure 7.1: Screenshot of a smart TV application showing related items as an overlay on top of a live TV news broadcast.

problem of linking encyclopedic content to a live television stream. Given the highly diverse set of information sources to be retrieved, including web pages, social media, and other video content, we focus in this chapter on information needs stemming from live television news broadcasts and provide an approach for retrieving archived video content.

To help see the rationale for our retrieval task, consider watching a news bulletin. As a news item is being covered, users may find themselves interested in watching background video material on one or more of the main entities or themes of the item, either to view at that very moment or to bookmark for later consumption. In order to present users with related video items that provide relevant background information, we require an algorithm that is able to automatically select video items for each news item as it is being broadcast. The output of a system implementing such an algorithm is depicted in Figure 7.1. Up to four related video items are suggested in an overlay on top of the news broadcast, each consisting of a keyframe and title; in the screenshot the left-most suggestion is highlighted (indicated by the red rectangle). The scenario that we cover in this chapter is one in which a user interrupts or pauses the newscast to find videos relevant to the current news item to learn more about the topic. The content retrieved to complement the ongoing broadcast is not personalized but should be related to the broadcast and interesting for a wide audience.

As explained in the previous chapter, in the setting of live television news, a constant stream of subtitles is typically available, generated with only a very slight delay as an aid for the hearing-impaired. To be able to retrieve related content, we analyze these subtitles and generate search queries from the stream that we subsequently use to search a video news archive. As a news item is being broadcast, additional textual information capturing the content of the news item becomes available. Thus, the challenge that we face is to iteratively incorporate this new information as it appears in the stream to generate and maintain an effective query model, and to do so in near real time.

In this chapter, we propose our solution: a *dynamic* query modeling approach that explicitly considers the dynamics of streaming sources and is suited for the near real-time setting of live TV. We cast the task of finding video content related to a live television broadcast as a Markov decision process (MDP) and use reinforcement learning to optimize the retrieval effectiveness of queries based on a stream of live television subtitles [17, 207, 223]. We significantly outperform state-of-the-art baselines for stationary query modeling [125] and for text-based retrieval in a television setting [25]. Thus, in this chapter, we make the following contributions:

(1) We formalize the task of related content finding to a live television broadcast as a Markov decision process.

(2) We propose a dynamic query modeling approach, using reinforcement learning to optimize a retrieval model that is sufficiently efficient to be run in near real time in a live television setting and that significantly improves retrieval effectiveness over state-of-the-art baselines for stationary query modeling and for text-based retrieval in a television setting.

(3) We provide a thorough evaluation and an analysis of when and why our approach works. We find that adding more weighted query terms and decaying term weights based on their recency significantly improve the effectiveness. Static term features derived from the target collection and background corpora are more important for selecting effective query terms than dynamic features that are dependent on the stream of subtitles.

The remainder of the chapter is organized as follows. To set the scene, related work is briefly summarized in Section 7.1 and discussed in detail above in Section 2.4. Section 7.2 describes our approach to dynamic query modeling. Our experimental setup is detailed in Section 7.3. Results are presented and analyzed in Section 7.4 and discussed further in Section 7.5. We conclude in Section 7.6.

7.1 Related Work

Dynamic query modeling is related to a number of tasks in information retrieval. We have discussed related work on search in a streaming setting, on query modeling, on temporal relevance feedback, and on reinforcement learning and MDPs in information retrieval in detail above in Section 2.4. Here, we briefly summarize this, and point out how our work differs.

Search in a streaming setting. We use dynamic query modeling to find related content based on a textual stream. In the previous chapter, we have studied the task of finding relevant background information for a second screen in a live TV setting. We have approached this as an entity linking problem and have explicitly modeled context. Although the tasks in this and the previous chapter both deal with streaming textual content, our work in this chapter differs because we model the entire textual stream and base our approach on modeling effective queries.

More closely related is the work of Henzinger et al. [96], who propose an approach to find relevant news articles during broadcast news. Every 15 seconds, they produce a two term query, using a “history feature,” consisting of the last three blocks of text. Recent work by Blanco et al. [25] on Yahoo! IntoNews builds and improves on the work of Henzinger et al. [96]. Their focus is on the problem of detecting a change of topic, for which they propose several segmentation approaches. After segmentation, queries are generated using a bag-of-word approach with TF.IDF scores for each term. We include the query modeling approach of Blanco et al. [25] as our first baseline.

Generic methods to automatically segment (e.g., TextTiling [95]) have been extensively studied and shown to perform well in streaming settings [6]. Specific segmentation approaches have been proposed for streaming settings similar to ours [25, 96]. Our dynamic query modeling approach can be applied after using these segmentation methods, instead of the relatively simple TF.IDF approaches used in [25, 96]. This would improve retrieval effectiveness, as we will see below in the comparison with the baseline due to Blanco et al. [25]. Furthermore, our approach of decaying term weights based on their recency might provide a combined solution for the segmentation and query modeling problems. In future work, we plan to further investigate the use of decaying term weights in a setting where we need to address segmentation. However, the subtitles that we work with are generated from an auto-cue and thus contain markings indicating the start and finish of an item. Segmenting items or switching from one news item to the next will therefore not be covered in this chapter.

Query modeling. A natural way of looking at search in streaming settings is to view it as a query modeling task, where the aim is to model an effective query based on a larger volume of text. This task has two distinct aspects: (1) query reduction and (2) content-based query suggestion. The former deals with reformulating long or descriptive queries to shorter and more effective queries. Driving this research is that shorter queries are not only more likely to retrieve more focused results, they are also more efficient to process [18]. The latter task is to generate queries and the methods used here are similar to those used for query reduction. Content-based query suggestion has been used for linking news archives across modalities [33] and for generating phrasal-concept queries in literature search with pseudo-relevant feedback [115]. The task of finding related articles that we considered in Chapter 3 could be seen as a content-based query suggestion task. We extended a query modeling approach with automatically extracted temporal references. Content-based query suggestion differs from content-based recommendation, where a user profile or user interactions are also available. For example, Bendersky et al. [20] propose a hybrid approach using content-based heuristics and signals from user interactions to retrieve video suggestions for related YouTube videos. Such a hybrid approach is not feasible in our scenario, as we have no user interactions that relate to the live TV content.

The state of the art in query modeling is formed by a family of related methods, e.g., [18, 19, 118, 124, 125]. The typical approach is to produce query terms, score each, and generate one or more queries. Term scoring is often based on term features taken from large background corpora. Queries are selected based on query difficulty prediction features; these require an initial retrieval round, and as such are not feasible in real time. For example, Lease et al. [125] generate queries of up to six terms by sampling term weights and then computing a metric to learn query term weights. Similar features are used in [19, 125, 126], with comparable retrieval scores. We propose a dynamic query modeling approach that is tailored for use in a streaming setting and outperforms the stationary baseline based on Lease et al. [125]. Our work differs in that we explicitly model the dynamic nature of streaming sources and in that we generate more complex queries (e.g., using field weights).

Reinforcement learning and MDPs. We model the task of finding video content related to a live television broadcast as a Markov decision process (MDP) [66] and base our approach on methods from reinforcement learning (RL) [207]. RL intertwines the concepts of optimal control and learning by trial and error. Central is the concept of an "agent" optimizing its actions by interacting with the environment. This is achieved by learning a *policy* that maps *states* to *actions*. An MDP is a specific type of reinforcement learning problem that was proposed before the field was known as reinforcement learning. In an MDP, we decide on the optimal action in a Markov process [17]. The Markov property holds when the policy for a state is independent on the previous states. A Markov state thus has to represent the entire history of previous states as far as this is relevant for the value of the policy.

As described above in Section 2.1.2, MDPs have been used to model diverse information retrieval (IR) problems, e.g., to explicitly model user behavior in session search [83, 132, 133]. In our setting, new subtitles keep coming in, hence we are dealing with a non-stationary MDP. More specifically, because the decision of what action to choose does not influence the states that emerge from the environment, this is considered an associative search task [15]. Perhaps the best studied application of reinforcement learning in IR is such an associative search task, online learning to rank [97, 177, 233]. Here, a retrieval system is optimized based on noisy feedback from user interactions. Our work differs from the work listed above in that we are not just optimizing rankings, but we focus primarily on query generation.

7.2 Dynamic Query Modeling

The search task we address is to find relevant content for a dynamic textual stream. We analyze the stream of subtitles that comes with television broadcasts and generate search queries from this stream. We subsequently use these to search a news video archive. Our dynamic query modeling (DQM) approach is designed to take streaming textual data as input and combines a retrieval model that defines a set of hyperparameters w with a learning approach that optimizes these hyperparameters w based on feedback. For learning, we regard the retrieval model as a black box with hyperparameters that affect the retrieval process and we obtain feedback based on the retrieval results. Our learning approach is therefore able to directly optimize for retrieval effectiveness. Figure 7.2 depicts the retrieval model and Figure 7.3 summarizes the learning approach. We discuss the learning approach in more detail in Section 7.2.3. The retrieval model and its hyperparameters will be further detailed below in Section 7.2.2, after we describe the terminology we will use.

7.2.1 Terminology

In Section 6.1, in the context of using subtitles for content linking, we defined dynamic textual streams as sources that continually produce "chunks" of text. A chunk is the amount of subtitle text that can be displayed on a single screen. Hence, chunks do not

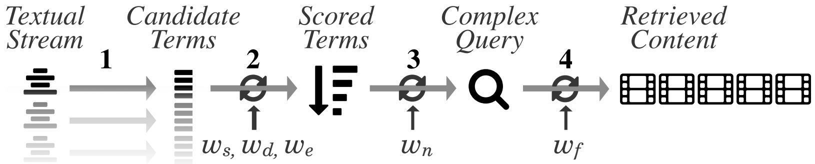


Figure 7.2: DQM retrieval model for textual streams, consisting of four steps: (1) query term candidate generation; (2) query term candidate scoring; (3) query generation; (4) retrieval.

necessarily form a grammatical sentence. However, as these chunks are produced to be read in sequence, syntactic phrases generally do not cross chunk boundaries. Chunks are relatively short, containing about seven terms on average. Chunks form a growing sequence $S = \langle s_1, s_2, \ldots \rangle$, where each s_i is a chunk. The task we address in this chapter is to generate, in real time, a query q_i for chunk s_i (having observed chunks $s_1, \ldots, s_{i-1}$) that is able to retrieve relevant video content at that point in the broadcast.

7.2.2 Retrieval Model

The retrieval model we employ consists of four major parts, which are depicted in Figure 7.2 and described in Algorithm 3. Our retrieval model consists of four steps. First, we incrementally update a list of candidate query terms that we obtain from the textual stream of subtitles. We then compute a score for each term with a weighted sum of term features. Next, we generate a query from the scored query term candidates. The generated queries are complex, consisting of many weighted query terms, and they are selected from the highest scoring query term candidates. Finally, we retrieve the results. The retrieval model defines a set of hyperparameters $w = w_d \cup w_s \cup w_f \cup w_n \cup w_e$ that each alter the retrieval process. The hyperparameters w_s and w_d can be construed as feature weights for the static and dynamic features (detailed below) and w_f as field weights, while hyperparameters w_n and w_e alter the number of selected query terms and the decay of query term candidate scores respectively. We provide more details with respect to these hyperparameters later in this section.

For the first step, *generating query term candidates*, we employ a bag-of-words approach which has proven to be effective in most retrieval settings, including settings similar to ours [125]. We generate a list of query term candidates by tokenizing the subtitles (line 2 in Algorithm 3). All terms from the subtitles in the video segment are considered as query term candidates. For each query term candidate we keep track of when we last saw this term (line 3).

The next step in the DQM retrieval model is to *assign a score* to each of the candidate query terms. A score for a query term candidate is initially computed as a weighted sum of the term's features. The feature weights are hyperparameters w_s and w_d for the retrieval model. We use a set of term features that are either static for the term or updated dynamically with the stream of text. The static features are computed once for each new query term candidate in the stream (line 4). The dynamic features are updated in each new state for all query term candidates (line 6). The dynamic and static features are listed in Tables 7.1 and 7.2 respectively.

Table 7.1: Dynamic term features for stream S.

$TF(t)$	Frequency of the term t in S
$TF.IDF(t)$	$TF(t)$, multiplied by $IDF(t)$ in the target collection
$AugmentedTF(t)$	$0.5 + \frac{0.5 * TF(t)}{max\{TF(t'):t' \in C\}}$
$Capitalized(t)$	Binary feature to indicate whether t has appeared capitalized in S

Table 7.2: Static term features and their per corpus availability.

		Target	Google	Wikipedia		
		collection	Web1T	title	body	anchors
$P(t,c)$	Probability of term t appearing in c		✓			
$TF(t,c)$	Frequency of term t in c	✓	✓			✓
$DF(t,c)$	Number of docs in c where t occurs	✓		✓	✓	✓
$IDF(t,c)$	log $\frac{\|c\|}{DF(t,c)}$, where $\|c\|$ is number of docs	✓		✓	✓	✓
$RIDF(t,c)$	Residual IDF [49]	✓				✓

The dynamic features are computed with information from the textual stream S. They include the term frequency $TF(t)$ and augmented term frequency, intended to prevent a bias towards longer documents. $TF.IDF(t)$ is computed using the target collection. The $Capitalized(t)$ feature indicates whether a term appeared capitalized in the stream, ignoring the beginning of a sentence. The intuition behind this is that a person or location is likely to appear capitalized and might be an effective query term.

We also compute five static features over three corpora (see Table 7.2). The first corpus we compute static features for is the target collection, where the features indicate whether the candidate query term will be effective in searching that collection. The two other corpora provide an indication of how common a term is across the web and in different document fields of an encyclopedic source. Both our subtitle source and the descriptions for the video items in our target collection are in Dutch. We therefore use the Dutch unigrams of the Google Web1T [28] corpus as counts of how common a term is on web pages. The counts were generated from approximately one hundred billion tokens. We use Wikipedia, with articles separated into title, body and anchor, as a source to indicate how common a term is in an encyclopedic text. Each should provide different clues about a term. A term appearing in the title might be more important than one appearing in the body. Likewise, if a term is used as anchor text, it might describe well what it refers to. Our five static features include the frequency of a term t, the number of documents it appears in and the probability of observing term t in the corpus c. From the document frequency we derive the inverse document frequency (IDF) and the residual IDF. This is the difference between the observed IDF and the value predicted by a Poisson model [49]: $RIDF(t,c) = IDF(t,c) - \log_2 \frac{1}{1-e^{TF(t,c)/|c|}}$, where $|c|$ is the number of documents in c. Not all features can be computed for each corpus, e.g., the Google Web1T collection lacks document counts. The sets of dynamic and static features are extended with logarithmically transformed values. All 38 resulting features are min-max normalized.

Algorithm 3: Dynamic query modeling

```
   Input: Textual stream $S = \langle s_1, \ldots \rangle$.
   Output: Stream of complex queries $Q = \langle q_1, \ldots \rangle$, where each query $q_i$ is a set of
           (term, field, weight) tuples.
   Data: Set of hyperparameters:
           $w_s$  weights of static features,
           $w_d$  weights of dynamic features,
           $w_e$  decay rate for a query term candidate weight,
           $w_n$  number of query terms for complex query,
           $w_f$  weight for each document $field$.
 1 foreach new chunk in $S$ do
       Step 1: Generate query term candidates
 2     Tokenize the new chunk.
 3     Update term candidates with new terms and last seen index.
       Step 2: Assign score to each query term candidate
 4     Compute static features for new candidates.
 5     foreach query term candidate do
 6         Compute dynamic features for each candidate term.
 7         Compute query term candidate score as the weighted sum of feature values,
           using weights $w_d$ and $w_s$.
 8         Decay query term candidate score with $e^{-w_e \cdot i}$, where $i$ is the relative age of
           term $t$ in the stream thus far (0 for the current chunk and 1 for the first chunk).
       Step 3: Generate complex query
 9     Select top $w_n$ query term candidates with highest score.
10     Generate complex query with individual field weights $w_f$.
```

For each query term candidate we compute a score as the weighted sum of all 38 features multiplied by an exponential term decay factor (lines 7 and 8 in Algorithm 3). This term decay factor is computed separately for each term candidate, based on how recently the term was observed in the stream S. The intuition behind this is that a more recently mentioned term might be more relevant than one that has not recently been mentioned. Hyperparameter w_e governs the decay rate. Concretely, we multiply the weighted sum of features with $e^{-w_e \cdot i}$, where i is the relative age of term t in the stream thus far. This relative age ranges from 0 to 1 and is 0 for the current chunk and 1 for the first chunk in the stream. The number of steps between 0 and 1 is equal to the number of chunks between the current and the first chunk in the stream.

The third step in the DQM retrieval model is to *generate a complex query* (lines 9 and 10 in Algorithm 3). From the ranked query term candidates we generate complex queries for the top n terms, where n is based on a hyperparameter w_n. The weights for each term in the resulting query are set to the score produced in the query term candidate scoring stage. We explicitly allow for negative weights for term features which may cause negative term candidate scores. In this case we omit the term, resulting in a query with

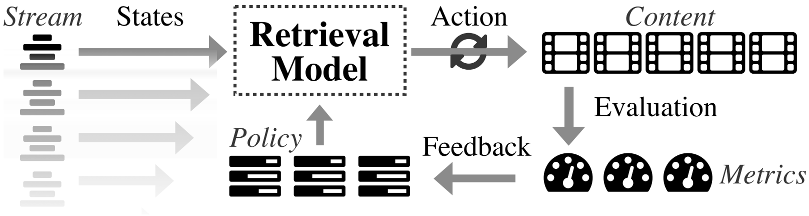


Figure 7.3: DQM learning approach. A textual stream is processed using a retrieval model that defines a set of hyperparameters. Feedback on the retrieval results is used to optimize the hyperparameters.

fewer query terms. We extend the query with learned field weights, allowing the model to learn to attribute more importance to specific fields, such as the title of a video. The field weights w_f are also exposed as hyperparameters.

The final step in the DQM retrieval model is to *retrieve* the results. For this we use a state-of-the-art search engine that uses language modeling for retrieval and is described further in Section 7.3.

7.2.3 Learning

The learning approach (depicted in Figure 7.3) considers the retrieval model as a black box that defines a set of hyperparameters, altering the retrieval process in a complex manner (cf. Section 7.2.2). We regard this learning problem as an MDP. A new *state* occurs when a new chunk of subtitles presents itself. In our setting we optimize the *action* of generating a query based on *feedback* in terms of retrieval effectiveness. We obtain this feedback by generating search results through our retrieval model that is governed by a set of hyperparameters. In reinforcement learning (RL) terms, we regard these hyperparameters as the *policy* that we are optimizing.

RL differs from supervised machine learning in that it explicitly deals with optimizing actions for the whole problem space based on goals within an uncertain environment. RL's trial and error type of learning fits well with the nature of our problem. We do not just optimize a single problem dimension, i.e., a combination of ranking features, but we optimize the entire process, from selecting and scoring candidate terms to generating a complex query.

We learn an optimal policy using the Dueling Bandits Gradient Descent (DBGD) algorithm [233], detailed in Algorithm 4. This algorithm iteratively tries a new policy w' that is a small change from the current best policy w. Based on feedback on the effects of both policies w' and w, it finds a winning policy. From this comparison it decides to keep the current policy w or to move the current best policy in the direction of w'. By iteratively updating the current best policy w, DBGD performs hill climbing to obtain an optimal policy based on the feedback. It thus obtains hyperparameter settings that optimize retrieval effectiveness for our DQM retrieval model.

DBGD has two parameters: step size δ for generating a new w' and learning rate α. Here, δ controls how big a change in weights we compare the current best to. The learning rate α controls how large a step is taken towards a better weight vector once it is

Algorithm 4: Our implementation of the Dueling Bandits Gradient Descent algorithm [233].

```
Input: step size $\delta$, learning rate $\alpha$, initial weights $w_1$, retrieval function $retrieve$, a
       $metric$, $maxIterations$
$w = w_1$
for $i \in \{1, \ldots, maxIterations\}$ do
    Sample small step $\epsilon$ of maximally length $\delta$.
    $w' = \frac{w+\epsilon}{|w+\epsilon|}$
    $a = retrieve(f * w)$
    $b = retrieve(f * w')$
    $d = metric(b) - metric(a)$
    if $d > 0$ then $w = \frac{w+\alpha*\epsilon}{|w+\alpha*\epsilon|}$
```

found. We initially set the weights w_1 randomly for the feature weights w_d and w_s and the number of terms w_n to 10. We initialize the retrieval model, with equal field weights w_f and a decay term w_d of 0.

To compare the effects of two policies w and w', we retrieve results produced by the DQM retrieval model for both policies. We are not in a position to experiment with actual users from whom we could obtain online feedback, e.g., using interleaved comparison methods [177]. We therefore use offline relevance assessments. For both retrieval results (obtained through w and w'), we compute a retrieval performance metric. This can be any metric; see Sections 7.3 and 7.4.3. Based on the outcomes, we decide whether the new policy w' is better than the current best policy w. In case of a draw, the current best is kept. If we were to use feedback from live user interactions, the feedback would be more noisy; we explore how noisy feedback affects learning in the analysis in Section 7.4.3.

Since the items of a news broadcast are segmented, our task can be performed naturally on subsequences, which we call "video segments" (i.e., *episodes* in RL terminology). The last chunk of a video segment produces a special state, the terminal state. It is followed by a reset to the start state. DQM is *episodic* with an *indefinite horizon*, i.e., episodes are finite, but without a predefined length. The policy we optimize is generic across episodes, i.e., it does not consider any episode-specific information and will not be reset after a terminal state. Our view of this learning problem as an episodic MDP fits well with the segmented nature of TV news.

A typical issue for any RL problem is deciding when to learn: should we explore new policy options or exploit the current policy. We assume that users are interested in related content near the end of an episode, as this is where they will have learned most about a topic. We therefore chose to only do an explorative learning action in the terminal state of an episode. In the other states, we exploit the current best policy. This way, we obtain less feedback but optimize the policy at the states where it is most important to perform optimal.

7.3 Experimental Setup

We seek to answer the following research questions regarding our proposed model and reinforcement learning approach:

RQ6 Can we effectively and efficiently find video content related to a live television broadcast in real time using a query modeling approach?

RQ6.1 Does dynamic query modeling improve retrieval effectiveness over state-of-the-art stationary query modeling baselines?

RQ6.2 What do the components of the DQM retrieval model contribute to the effectiveness?

RQ6.3 How do the reinforcement learning parameters, choice of optimization metric and noisy feedback influence the speed of the learning process and the resulting retrieval effectiveness?

To answer these research questions, we create an annotated dataset and set up experiments. Below, we describe the dataset, experimental set-up and detail our evaluation.

Subtitles. We obtain a dataset of subtitles for the hearing impaired from the Dutch eight o'clock evening news broadcast of the Nederlandse Omroep Stichting (NOS), the Dutch public news broadcaster. We selected this as our source because its content is diverse and volatile; it may cover items broadly and for minutes, or just very briefly. A news broadcast typically lasts about 25 minutes and contains around ten items, which we refer to as *video segments* or *episodes* in RL terms. A typical video segment consists of 44 chunks of on average seven terms per chunk and 306 terms per video segment. For evaluation purposes, seven news broadcasts are randomly selected from broadcasts dated May 2013, containing 50 video segments in total. The video segments do not overlap in main topic. The subtitles are prepared based on the text for the teleprompter or auto-cue. It therefore contains markings to indicate when a new video segment starts.

Collection. As our target collection we use the video archive of the same news broadcaster, NOS, which contains individual news item videos. The video items are often taken from news broadcasts, but can also be longer versions of interviews or aimed to provide further background. This collection is publicly available and can be crawled via their website.[1] Our index covers the years 2011–2013 and contains $37,884$ video items, with an average of 40 video items per day. For our experiments, we limit the queries to only the video items that were published before the news broadcast. We consider the title, description and tags as different textual fields. Note that we do not use any video specific information and thus regard the video items simply as textual metadata records.

Ground truth. To establish ground truth for our evaluation,[2] we ask assessors to read the subtitles of a video segment and then rate videos for relevance. These items are obtained by pooling the top rankings for each baseline to on average 79 videos per segment. The video items in the pool are presented in random order. We train two assessors and each

[1] `http://nos.nl`
[2] `http://ilps.science.uva.nl/resources/sigir2015-dqm`

one annotated half of the video segments. Our instructions to the assessors were: *Imagine that you interrupted the news broadcast after the segment because you're interested in watching related video content. How would you rate each video?* We use a five-point scale to capture the level of relevance of each video [45] (with the usual labels *perfect*, *excellent*, *good*, *fair* and *bad*). The distribution of the ratings over the respective labels from perfect to bad is 5%, 5%, 8%, 16% and 66%. This suggests that the task is not an easy task, but there are plenty of good videos to rank. For each video item, the assessors are provided with all metadata (title, date, description, keywords) and can watch and explore the original video to make a better judgement.[3]

Evaluation. In our setting, a viewer is searching for related video content for a news broadcast item, most likely at the end of an item or just after it has finished. We therefore evaluate the retrieval effectiveness at the end of a video segment or in RL terms, the terminal state of an episode. In the DQM setting, it is important to provide the most relevant video and to provide them as high in the ranking as possible. In a live TV setting, a user would not examine a full result list, but only a limited number of video items. As our main evaluation metric we choose normalized discounted cumulative gain (nDCG) [104], as it can handle graded relevance assessments and takes positional importance into account. We compute nDCG for the entire result list and for the first five positions as nDCG@5, skipping video items that were not annotated. We perform leave-one-out cross validation to evaluate our approach. For each video segment we train an individual model where all other video segments serve as training material. We then evaluate the effectiveness on the video segment that was left out of the training set. We consider nDCG@5 more relevant for our setting but optimize for nDCG as it is smoother than nDCG@5 and already gives high importance to the retrieved documents at the top of the ranking. We assume that taking the improvements in ranking below the fifth position into account will benefit learning in the long run. We revisit this decision in the analysis of our approach (Section 7.4).

Retrieval engine. For all experiments we use the language modeling and inference network-based Indri retrieval model [146], with stopword removal and without stemming. We use Dirichlet smoothing with smoothing parameter μ set to the default 2500.

Baselines. Since we evaluate at the end of a video segment, we can compare to stationary *baselines*. For these baselines, we concatenate all subtitles of a video segment to form a single pseudo document. Based on this pseudo document we search for related video content. In this way, the task becomes similar to the more-like-this task, that is supported by many search engines. We include two stationary approaches, that we label "Baseline" and "Modified Lease." The first, "Baseline", uses the top-10 terms from a bag-of-words model of the pseudo document, ranked by TF.IDF score (where the document frequency is computed on the target collection). This baseline is how Blanco et al. [25] perform query modeling in their text-based retrieval approach for TV. The second stationary approach ("Modified Lease") is comparable to the state-of-the-art model of Lease et al. [125]. The retrieval model is based on regression to learn queries that consist of no more than six terms. The terms are selected and weighted based on supervised machine learning (regularized linear regression) using a bag-of-words representation of the pseudo document. The

[3] We evaluate assessor-agreement over a set of 25 videos from five doubly annotated segments; Spearman's rank correlation measures 0.8636, signaling good agreement.

Table 7.3: Retrieval effectiveness of the dynamic query modeling (DQM) approach vs the two baselines. Significant differences, tested using a two-tailed paired t-test, are indicated with $^{-}$ (none), $^{\triangle}$ ($p < 0.05$) and $^{\blacktriangle}$ ($p < 0.01$); the position of the symbol indicates whether the comparison is against row 1 (left most), row 2 (center) or row 3 (right most).

Method	**nDCG@5**	**nDCG**
1. Baseline [25]	0.6486$^{\ \ -\blacktriangledown}$	0.6113$^{\ \ -\blacktriangledown}$
2. Modified Lease [125]	0.6994$^{-\ \ -}$	0.6484$^{-\ \ \blacktriangledown}$
3. DQM^{-}	0.7570$^{\blacktriangle -}$	0.7393$^{\blacktriangle\blacktriangle}$
4. DQM^{-} + field weights	0.7651$^{\blacktriangle\triangle -}$	0.7566$^{\blacktriangle\blacktriangle\triangle}$
5. DQM^{-} + term weighting	0.7814$^{\blacktriangle\blacktriangle -}$	0.7845$^{\blacktriangle\blacktriangle\blacktriangle}$
6. DQM^{-} + term and field weighting	0.7781$^{\blacktriangle\blacktriangle -}$	0.7945$^{\blacktriangle\blacktriangle\blacktriangle}$
7. DQM^{-} + decayed term weighting	0.7940$^{\blacktriangle\blacktriangle\blacktriangle}$	0.7897$^{\blacktriangle\blacktriangle\blacktriangle}$
8. DQM	0.8005$^{\blacktriangle\blacktriangle\blacktriangle}$	0.8072$^{\blacktriangle\blacktriangle\blacktriangle}$

features we use for the Modified Lease baseline are the same as for our DQM approach, excluding the features based on Wikipedia (not used in [125]). Furthermore, we choose not to include the simple part-of-speech features and the lexical context features (the word before and word after) from their model. These features get relatively low weights in [125] and are less applicable in our setting than in their descriptive queries setting (hence, we dub this approach "Modified Lease").

7.4 Results & Analysis

We describe the results of our experiments and investigate the effectiveness of our DQM approach, following the three research questions listed in the previous section.

7.4.1 Retrieval Effectiveness

To answer RQ6.1, we compare the effectiveness of DQM to that of the stationary baselines. Table 7.3 shows the performance in terms of retrieval effectiveness. Both baselines show a decent performance with an nDCG score of around 0.7. We cannot directly compare the performance of the baselines to that of Lease et al. [125], as they report on different collections and use shorter descriptive queries. Surprisingly, the modified Lease approach is not able to significantly improve over the less complex baseline. A plausible explanation for this is the limited number of query terms in the modified Lease approach; see the analysis in Section 7.4.2 below.

Rows 3–8 of Table 7.3 show the retrieval effectiveness of our DQM approach, building up from a basic approach (labeled DQM^{-}) to the full DQM approach. DQM^{-} uses only the dynamic and static term features to select query terms. It is, however, already able to significantly improve in terms of nDCG over both baselines (row 3). Next, we look at the weighting in DQM. Enabling field weighting gives a small boost in retrieval effectiveness (row 4). If we enable term weighting based on the machine learned scores, the model performs significantly better on the full result list, although the effect is not significant

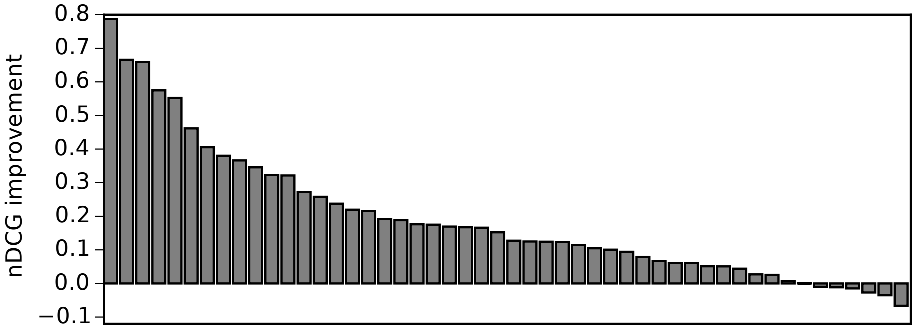


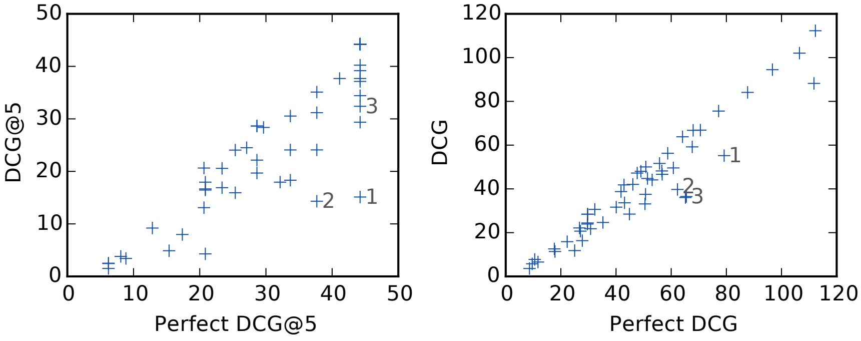
Figure 7.4: Improvement in terms of nDCG for DQM over the baseline, where each bar represents a video segment.

on nDCG@5 (row 5). Interestingly, adding field weights to the term weighting approach results in a drop of retrieval effectiveness for the top five, but improved effectiveness for the full rank list (row 6). Enabling term weight decay without field weights gives an additional boost towards a significantly better approach than the base approach, both on the top five and the full result list (row 7). Finally, our full DQM approach in row 8 significantly improves over the stationary baselines and the base DQM$^-$ approach. The effects are similar in nDCG and nDCG@5, although they are clearer for nDCG, i.e., the full ranked list.

To investigate where our DQM approach works, we look at the effectiveness per video segment compared to the baseline. We plot the difference in nDCG in Figure 7.4. Our approach is able to substantially improve retrieval effectiveness for the bulk of the video segments. There are seven segments where our approach hurts performance. A closer look at these reveals that they mostly already have a high nDCG score for the baseline. There does not seem to be an influence of the length of the segments.

We further study the effectiveness across video segments by separating the nDCG scores into their components: the DCG score and the perfect DCG score. *Perfect DCG* is the DCG score obtained when the documents are ideally ranked according to the ground truth. If a ranking is ideal, the DCG score is equal to the perfect DCG score and nDCG is equal to 1. We plot the DCG scores versus the perfect DCG scores in Figure 7.5, for the top five and the full result list. A perfect nDCG score would be on the diagonal. The closer a result is to the bottom, the lower the nDCG score. We see a similar pattern for the full ranked list as for the top five, although more distinctly for the top five. In some cases, DQM is able to obtain perfect scores for video segments with both high and low perfect DCG scores. However, it also performs below perfect in other cases; this does not seem to be related to the perfect score that can be obtained.

We look in more detail at three video segments that are far from the diagonal (and thus have a low nDCG score), marked in Figure 7.5 as 1, 2 and 3. Interestingly, segments 1 and 3 are in the top five most improved by our approach and segment 2 is one of the few that is hurt by DQM (respectively the 4th and the 5th bar from the left and the 4th from the right in Figure 7.4). Segment 1 is broad, linking French protests against a new law on gay marriage to a movie at the Cannes film festival. Segment 3 is a short item



(a) Top five

(b) Full result list

Figure 7.5: DCG for DQM versus perfect DCG for each video segment for the top five (left) and for the full result list (right).

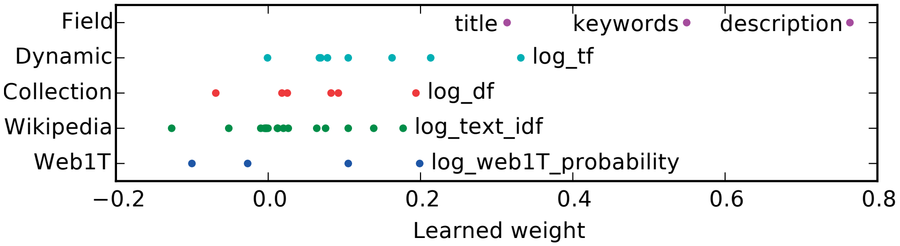


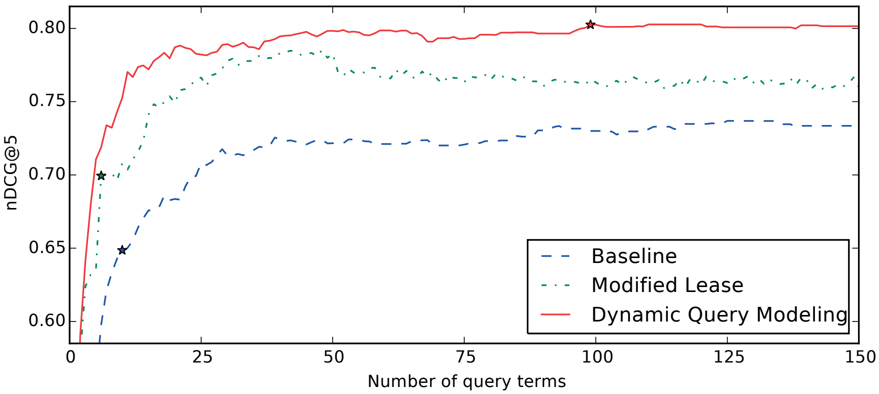
Figure 7.6: Feature and field weights for the DQM approach. The field weights and top weights for each group are annotated.

about the increase in ATM robberies and segment 2 is a broad item about the relationship between the Netherlands and Germany, on the occasion of a trade summit. From this we can see that, although DQM gets substantial improvements for broad items, there is still something to be gained.

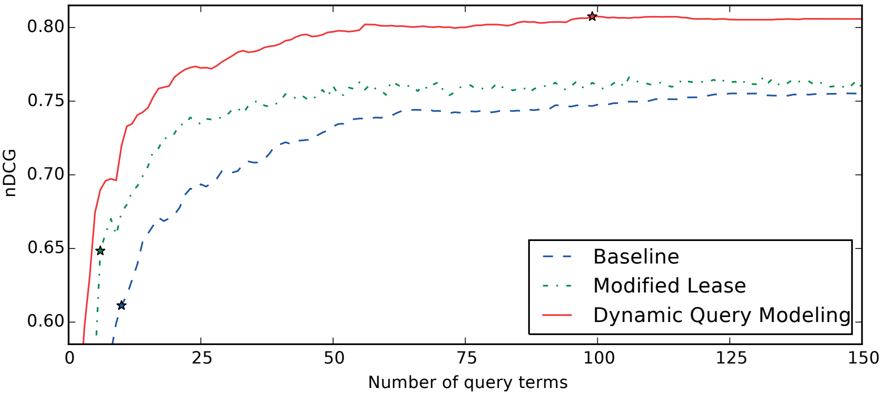
7.4.2 Analysis of Components

To answer RQ6.2, we investigate the contribution of the components of our approach. The learned values of the DQM retrieval model hyperparameters are plotted in Figure 7.6. Looking at field weights w_f, we observe that different values are learned, the highest weight is twice the value of the lowest weight. The highest weight is given to the longer description field. The keywords get a higher weight than the title field.

Number of weighted query terms. Next, we turn to the number of weighted query terms that DQM generates. In our experiments, this parameter was set through RL to a value close to 100 terms. This can include terms with very small weights or even negative weights (and thus not included in the final query). This is a substantially larger number than the fixed number of 6 query terms in the Lease approach. We investigate the impact on the retrieval effectiveness of DQM and the baselines; see Figure 7.7. From



(a) nDCG@5



(b) nDCG

Figure 7.7: Baseline retrieval effectiveness with different number of query terms as measured on the top five (nDCG@5, top) and on the entire result list (nDCG, bottom). The settings used in our main experiments are marked with a star.

this figure we observe that both baselines clearly benefit from adding more terms to the query. The modified Lease approach consistently outperforms the baseline approach. Interestingly, for the baseline approach, adding more query terms than around 30 does not further improve the effectiveness on the top five results, but does steadily improve the effectiveness on the entire ranked list. The modified Lease approach actually shows a drop in effectiveness on the top five when using more than around 50 query terms. For any number of query terms, DQM consistently outperforms both the baseline and Modified Lease on both metrics. For DQM, the retrieval effectiveness in terms of nDCG for the top five results and the entire result list show a qualitatively similar pattern. The effectiveness of DQM increases with the number of query terms, until it reaches a plateau at around 40 query terms.

Table 7.4: Ablation analysis of the contributions of DQM term feature sets to retrieval effectiveness. Significant differences, tested using a two-tailed paired t-test against row 1, are indicated for rows 2–9 with $^{-}$ (none), $^{\triangledown}$ ($p < 0.05$) and $^{\blacktriangledown}$ ($p < 0.01$).

Method	nDCG@5	nDCG
1. Full Dynamic Query Modeling (DQM)	0.8005	0.8072
2. without static features	0.6884$^{\blacktriangledown}$	0.6721$^{\blacktriangledown}$
3. without dynamic features	0.7823^{-}	0.7698$^{\blacktriangledown}$
4. without Web1T features	0.8012^{-}	0.7967$^{\triangledown}$
5. without Wikipedia features	0.7459$^{\blacktriangledown}$	0.7485$^{\blacktriangledown}$
6. without Collection features	0.7421$^{\blacktriangledown}$	0.7267$^{\blacktriangledown}$
7. with only Web1T features	0.7387$^{\blacktriangledown}$	0.7377$^{\blacktriangledown}$
8. with only Wikipedia features	0.7633$^{\triangledown}$	0.7617$^{\blacktriangledown}$
9. with only Collection features	0.7571$^{\triangledown}$	0.7583$^{\blacktriangledown}$

Term decay. In our experiments the term decay factor w_e was set through RL to a value of 0.5601. Our retrieval model thus has a mild preference for more recent terms. With this decay factor query term candidates from the first chunk are given a score that is 57% lower than the most recent terms. In Table 7.3, we observed a small boost in retrieval effectiveness when enabling decayed term weights for our full DQM approach. Interestingly, adding term decay to our baseline gives an nDCG improvement of 11.90% for the top 5 and 16.29% for the full ranked list. This suggests that term decay does indeed contribute to improving retrieval effectiveness.

Term feature weights. Next, we turn to the term feature weights w_d and w_s, plotted in Figure 7.6. We observe that a diverse set of features receive high weights; there is no single most important feature. The highest weights are assigned to the dynamic features. Of the static features, the collection features receive the highest weights, specifically the logarithm of the document frequency. For Wikipedia-based features, the log IDF in text receives the highest weight. The weights for features derived from the title and anchor text are substantially smaller than for the all Wikipedia text. For Web1T, the log term probability receives the highest weight. Five features receive negative weights, most notably the document frequency in Wikipedia text and the frequency in Web1T. These weights indicate a natural negative bias for common terms.

Contributions of components. We perform an ablation study and disable parts of DQM to investigate the effects on the retrieval effectiveness; see Table 7.4. We evaluate the effectiveness of our DQM approach with specific sets of term features disabled. These results are presented in rows 2–6 of Table 7.4. On rows 2 and 3, we see the effects of disabling respectively the entire set of dynamic and of static term features. Both result in a significant drop of performance. In fact, if we disable all static term features, the performance on the top five drops below the modified Lease baseline. Clearly, the static term features are essential for the performance of DQM. This makes it surprising that the dynamic term features obtain the highest weights. To further investigate this, we disable specific subsets of static term features. We observe from rows 4–6 in Table 7.4 that disabling any of the subsets significantly degrades the performance of DQM. For Web1T, this effect is not significant in the top five and not as significant as for the Wikipedia and collection features. Lastly, we investigate whether our strong static term features are strong enough by themselves, without the other features. The results in rows 7–9 of Table 7.4 show a qualitatively similar pattern to rows 4–6. Using only one subset of the static term features and no dynamic features significantly degrades the performance, although all variants still outperform our baselines. Using only Wikipedia for term features comes closest to our full DQM model, but it still performs significantly worse.

7.4.3 Learning

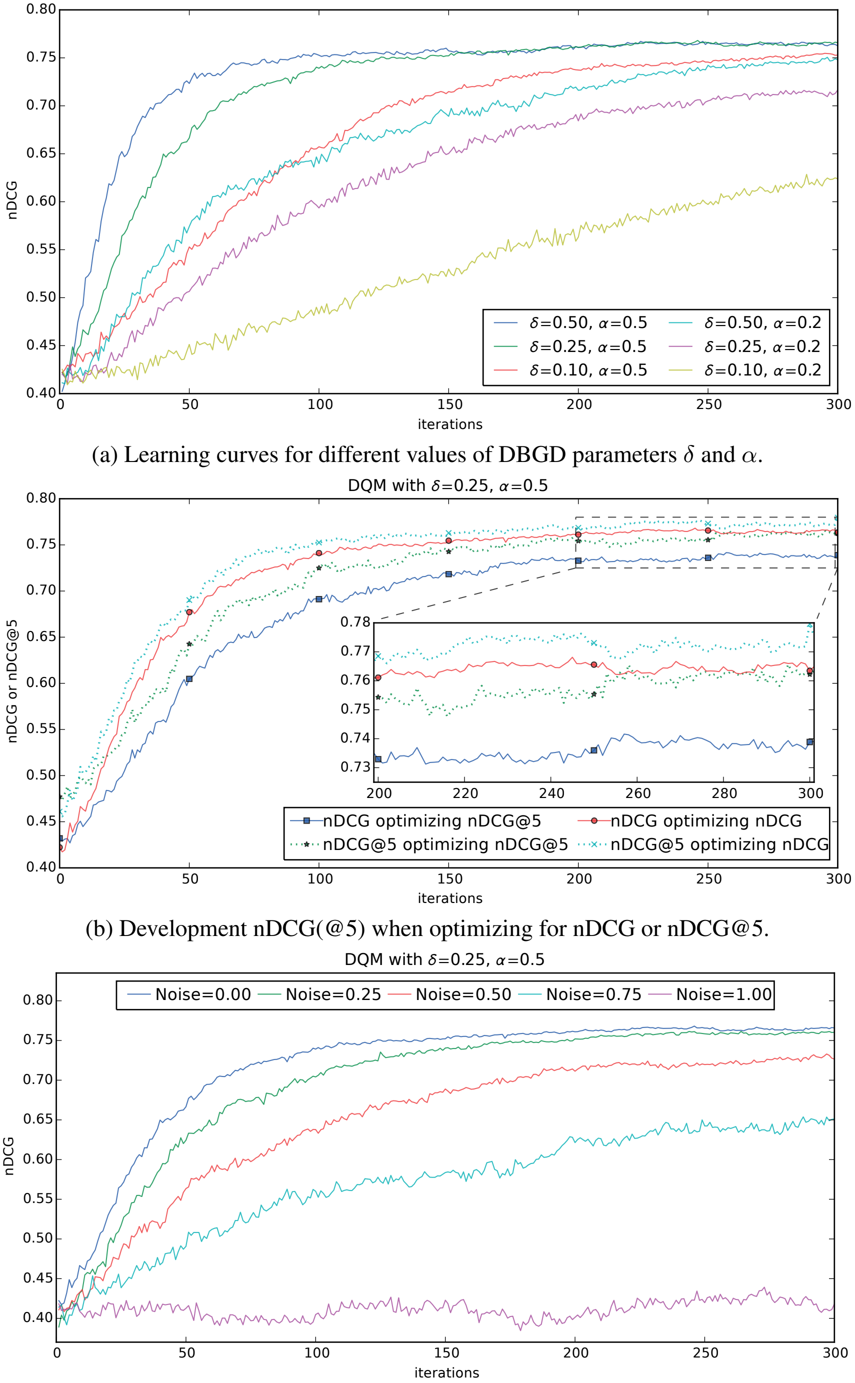
To answer the reinforcement learning related research question RQ6.3, we investigate how DQM learns by looking at learning curves. These curves are generated by evaluating the DQM model at each iteration on the left-out video segments. The curves are averaged over five runs for each of the 50 video segments and are thus comparable to the numbers in Tables 7.3 and 7.4.

Reinforcement learning parameters. First, we turn to the two parameters of the DBGD algorithm: the step size δ and the learning rate α. δ determines the distance between the current best weights and the new weights. The learning rate α is the size of the step taken towards the best scoring weight vector. We explore different values for δ and α and plot the results in Figure 7.8a.

We can observe from Figure 7.8a that for the same step size δ, a larger learning rate means that we reach higher effectiveness in fewer iterations. A larger step size δ will also result in faster learning, although there appears to be a risk in setting the δ to high. Despite early gains when using $\delta = 0.5$ and $\alpha = 0.2$, at around 75 iterations, it gets taken over by the run with the much smaller step size of $\delta = 0.1$ and higher learning rate of $\alpha = 0.5$. At $\alpha = 0.5$ the medium step size $\delta = 0.25$ appears to be able to better learn the subtleties when the effectiveness reaches a plateau, in comparison with the higher learning rate $\alpha = 0.5$.

Next, we investigate how optimizing for nDCG influences the performance as measured by nDCG@5. In Figure 7.8b we plot nDCG and nDCG@5 for runs in which we optimize either the nDCG or the nDCG@5 metric. We observe a consistently higher nDCG value when optimizing nDCG versus optimizing nDCG@5. Interestingly, we also see a consistently higher nDCG@5 value, confirming our idea that optimizing nDCG will also optimize nDCG@5.

Noisy feedback. DQM is also suited for use in a setting where feedback comes from user interaction and thus is not always perfect. We study how well it can deal with noisy feedback. For this we run a variant of DQM where we replace the comparison in the DBGD algorithm with a noisy comparison. This noisy comparison randomly returns a random comparison outcome. The noise level controls how often this occurs, where a noise level of zero is equal to our regular setting and a noise level of one results in completely random feedback. The results of this analysis are presented in Figure 7.8c. We observe that given completely random feedback, DQM will not improve in terms of effectiveness. However, it will also not degrade in performance. Adding more noise to the feedback for DQM will increase the number of iterations it takes to learn an optimal



(a) Learning curves for different values of DBGD parameters δ and α.

(b) Development nDCG(@5) when optimizing for nDCG or nDCG@5.

(c) Influence of noise on the convergence rate of effectiveness.

Figure 7.8: Learning curves showing the development in terms of effectiveness for the DQM approach across learning iterations. Metrics are computed using leave-one-out cross validation and averaged over five runs and 50 video segments at each iteration.

value, but an optimal value will be found even with highly noisy feedback. This suggests that DQM is also suited for use in a setting where feedback comes from user interaction.

7.5 Discussion

We see two obvious improvements to retrieval effectiveness for our DQM approach, which we will discuss here. First, we could use pseudo-relevance feedback (PRF) to do query expansion, an approach shown to be effective for generating phrasal-concept queries in literature search [115]. To increase recall, the original query is expanded with pseudo-relevant terms taken from the top retrieved documents. We experimented with enabling pseudo-relevant feedback. For the baseline, this increases nDCG with 3.5% for the top five and 7.3% for the full ranked list. We observed no improvement in retrieval effectiveness for DQM when enabling PRF. A possible explanation is that DQM's retrieval model uses many query terms, selected partly based on term statistics for the target collection. Instead of Indri's out-of-the-box PRF approach, a PRF approach that is tailored to our DQM retrieval model and the complex queries that it produces might be able to improve retrieval effectiveness.

Second, a plausible assumption is that more recent videos might be more relevant. We therefore also explored using temporal document priors, specifically ones inspired by human memory [173]. The DQM learning approach found no preference for recent documents and we observed no improvements in retrieval effectiveness with a temporal document prior compared to without one. An analysis of our ground truth confirmed that more recent documents were not likely to be more relevant. This might be due to our retrospective annotation of the ground truth. In a live TV setting, with feedback from actual user interactions, more experimentation with a temporal document prior (e.g., a Weibull decay) would be advised.

So far, the discussion of our results has focused on effectiveness. However, our target setting is live TV, where efficiency is of great importance. To verify that we can efficiently perform DQM in near real time, we compute the average number of chunks we process per second on a single core machine, averaged over ten passes over all 50 video segments. DQM is able to process 23.9 chunks per second. In Section 6.3, we reported that we observe an average of 0.24 chunks per second in a live TV setting similar to the one considered here, two orders of magnitude less than what we are able to process.

7.6 Conclusion

In this chapter, we have formalized the task of finding video content related to a live television broadcast as a Markov decision process and have proposed a reinforcement learning approach to directly optimize retrieval effectiveness, in order to answer:

RQ6 Can we effectively and efficiently find video content related to a live television broadcast in real time using a query modeling approach?

We have shown that our approach significantly improves retrieval effectiveness over state-of-the-art stationary baselines, while remaining sufficiently efficient to be used in near real time in a live television setting. We have shown that each DQM retrieval model component contributes to the overall effectiveness. A larger number of weighted query terms significantly improve effectiveness. Static term features that are dependent on the target collection and background corpora are more important for selecting effective query terms than dynamic features derived from the stream of subtitles. Decaying term weights based on their recency further improves retrieval effectiveness.

Regarding our reinforcement learning approach we have found that a medium explorative step size and a larger learning rate are the best choice in our setting. We have shown that optimizing nDCG also yields the best nDCG@5 scores. Lastly, we have shown that our reinforcement learning based approach to DQM still learns effectively when feedback becomes noisy. This suggests that our DQM approach is also suited for use in a setting where feedback comes from user interaction.

To understand the broader applicability of our work, it helps to point out that the task of finding related content to a live television broadcast combines two traditional basic information retrieval tasks: search and filtering [16]. In a typical search task the query changes and the collection remains static. In a typical document filtering task, a standing query is used to filter a stream of documents. Our task concerns both. Similar tasks exist, such as summarizing social media in real time [182] and finding replications of news articles while they appear [212]. We believe that our DQM approach is applicable to those tasks too.

As to future work, we plan to explore learning from noisy feedback from actual user interactions. The DQM algorithm is designed with such feedback in mind. We have shown that we can deal with synthetic noisy feedback and still learn to generate effective queries when feedback becomes noisy. It would be interesting to see if this is also the case with actual noisy feedback from users. An open question is how to interpret the user feedback, e.g., is a click always a positive signal?

Our dynamic query modeling algorithm generates a single query at any point in time. Future work could address design an extension that generates multiple queries and merges either queries or results. A contrastive experiment can show whether this is more effective. Similarly, extensions that personalize or explicitly generate temporally or topically diverse results may enhance the user experience. We could model this as a slot filling problem, where we have four video slots for which we select the most interesting videos to show to a user. Our evaluation considered this task a ranking task, but the slot filling problem fits closer with the actual user experience. The question is if such a view of the problem would lead to a different solution.

8 Conclusions

In this thesis, we presented work towards a core aim of information retrieval (IR): providing users with easy access to information. Three research themes have guided the research presented in this thesis, touching on three aspects of IR research: the *domain* in which an IR system is used, the *users* interacting with the system, and the different access *scenarios* in which these users engage with an IR system. Central to these research themes has been the aim to gain insights and develop algorithms to support searchers in their quest, whether it is a researcher exploring or studying a large collection, a web searcher struggling to find something, or a television viewer searching for related content.

The first research theme of this thesis was motivated by how researchers explore and study large collections. Inspired by their information seeking tasks, we have proposed computational methods to connect collections and for inferring the perspective offered in a news story. We have shown that we can effectively extract temporal references from collections and leverage these for the task of retrieving related articles. Furthermore, we approach human performance on the task of frame detection in news. To illustrate how researchers can benefit from our new algorithmic approaches, we have presented novel exploratory search interfaces and use cases. Our findings have implications for the research methodologies commonly employed in these fields and we have described approaches to adapting these.

For the second research theme, we have characterized how web searchers behave when they cannot find what they are looking for. We have shown significant behavioral differences given task success and failure. We have used our findings to propose ways in which systems can help searchers reduce struggling. Key components to support searchers are algorithms that accurately predict the nature of future actions and their anticipated impact on search outcomes. We have presented algorithms and experimental results for such components. Our findings have implications for the design of search systems that help searchers struggle less and succeed more.

In the third and final research theme, we considered a pro-active search scenario, specifically in a live television setting, where we have proposed algorithms that leverage contextual information to retrieve diverse related content for a leaned-back TV viewer. These algorithms can effectively and efficiency retrieve content related to a developing news story in near real time. Our research opens up new applications for IR technology in a live TV setting.

We have covered research on diverse IR tasks within three research themes. For each research theme, we have looked at the behavior of a user in a specific domain and scenario and proposed new algorithms for that user, domain and scenario. Returning to our aim of developing algorithms and gaining insights to improve the ease of access to information for users of IR systems, we conclude that each research chapter in this thesis has produced insights and algorithms that help ease some of the pains that searchers experience while using IR applications. At the end of this chapter, we provide an outlook on future research directions that the work presented in this thesis has opened up. First, we give a more detailed summary of the contributions and results of the research chapters, and answer the research questions set out at the beginning of this thesis. Both sections are organized along the three research themes.

8.1 Main Findings

Here, we summarize our main findings in more detail along the three research themes and six research questions presented in Section 1.1.

Theme 1—Studying News Collections
Within this research theme we have asked and answered two research questions. We started our investigation by focusing on how researchers explore and study large collections, and in particular on how they study news articles in their context. As a motivational use case, we have presented an exploratory search interface for a large news collection in which we use visualizations of the entire collection and of specific parts of the collection. We have investigated how researchers from the humanities select subsets of large collections for close examining. This document selection process is a combination of three of the four information seeking tasks historians employ according to Duff and Johnson [59]: orientation, known material search and relevant material identification. We have described how exploratory search and text mining techniques fit within these tasks.

We studied in more detail the information seeking task of building contextual knowledge [59] to ground further research. Humanities scholars critically regard historical sources in their context considering aspects such as date, authorship and localization in order to assess the credibility of a source [78]. When multiple sources are considered, each might provide an interestingly different perspectives on a historical event, expressing views at the time of writing or even a subjective view of the author. Where news articles have a clear temporal footprint, other sources such as encyclopedic articles might not. For these, temporal references can be extracted and leveraged in combination with the textual content to find related articles. We have cast this as an IR task and have asked:

RQ1 Can we effectively extract temporal references from digitized and digital collection and does leveraging these temporal references improve the effectiveness of retrieving related articles?

In answer to RQ1, we found that we can extract temporal references effectively in digital and digitized collections. We found that digitized collections pose interesting challenges requiring improved preprocessing. Leveraging these extracted temporal references improved effectiveness on the task of retrieving related articles. To illustrate how our algorithms could be used to automatically create connections between digital and digitized collections, we have introduced a novel search interface to explore and analyze the connected collections that highlights different perspectives and requires little domain knowledge.

Next we turn our attention to social scientists and the central question of content analysis: “Who says what, to whom, why, to what extent and with what effect?” [121]. We studied in more detail how news stories are framed, i.e., the way in which journalists depict an issue in terms of a ‘central organizing idea’ [76]. Frames can be seen as a perspective on an issue. Complex characteristics of messages such as frames have been studied using thematic content analysis [202]. Indicator questions are formulated, which are then manually coded by humans after reading a text and combined into a characterization of the message. To scale frame analysis up to large collections, we operationalized this as a classification task and asked the following research question:

RQ2 Can we approach human performance on the task of frame detection in newspaper articles by following the way-of-working of media analysts?

To answer RQ2, we followed the way-of-working of media analysts and proposed a two-stage approach, where we first rate a news article using indicator questions for a frame and then use the outcomes to predict whether a frame is present. Our ensemble-based approach to directly predicting the presence of a frame was the most effective and improved substantially over the approach that mirrors the manual approach. We found that for the task of frame classification, explicitly modeling the manual thematic content analysis does not improve performance. Our ensemble-based direct classification approach proved sufficient to capture the complex characteristics of frames that the indicator questions are aimed to represent. The results of an analysis using a model that explicitly models coder bias and the relatively low inter-coder agreement suggested that coders have different interpretations of the indicator questions for the frames. Like the indicator questions that represent different aspects of complex characteristics of messages, it seems that human coders represent different views on these aspects and characteristics.

Theme 2—Struggling and Success in Web Search

In the second research theme of this thesis, we turned our attention to struggling behavior in web search. When searchers struggle to find relevant information, this leads to frustrating and dissatisfying search experiences, even if they ultimately meet their search objectives. We addressed this important issue using a mixed methods study using large-scale logs, crowd-sourced labeling, and predictive modeling and asked two research questions, starting with:

RQ3 How do web searchers behave when they cannot find what they are looking for?

Through a log analysis of millions of search tasks, we have characterized aspects of how searchers struggle and (in some cases) ultimately succeed. We have found that struggling searchers issue fewer queries in successful tasks than in unsuccessful ones. In addition, queries are shorter, fewer results are clicked and the query reformulations indicate that searchers have more trouble choosing the correct vocabulary. We have shown significant behavioral differences given task success and failure. We therefore asked:

RQ4 How do web searchers go from struggle to success and how can we help them make this transition?

To answer RQ4, we developed and applied a crowd-sourced labeling methodology to better understand the struggling process and where it became clear the search would succeed. This pivotal query is often the last query and not all strategies are as likely to be pivotal. We developed classifiers to accurately predict key aspects of inter-query transitions for struggling searches, with a view to helping searchers struggle less. We have used our findings to propose ways in which systems can help searchers reduce struggling. Key components of such support are algorithms that accurately predict the nature of future actions and their anticipated impact on search outcomes.

Theme 3—Pro-active Search for Live TV

Motivated by the observation that less successful searchers experience more difficulty selecting the correct query vocabulary, we continued the research in this thesis in a pro-active search setting. Here, we try to automatically generate queries and find relevant content for a user, based on the context of their search. Motivated by the rise in interactive television and so-called second screen applications we introduced a new task: real-time entity linking of streaming text and asked:

RQ5 Can we effectively and efficiently provide background information for a live television broadcast in real time using an entity linking approach and does explicitly modeling streaming context improve the effectiveness?

To answer RQ5, we have created a dataset for this task and have shown that learning to rerank can be applied to significantly improve an already competitive retrieval baseline and that this can be done in real time. Careful pruning and leaving out several features that are heavy to compute does not significantly decrease effectiveness, while substantially increasing efficiency. Additionally, we have shown that by modeling context explicitly we can significantly improve the effectiveness of this learning to rerank approach. The proposed retrieval-based and graph-based methods to keep track of context are especially well-suited for streaming text, as we can incrementally update the context model. This makes these context models suitable for real-time applications.

In a similar setting, we considered the task of finding video content related to a live television broadcast for which we leverage the textual stream of subtitles associated with the broadcast. We asked the following research questions:

RQ6 Can we effectively and efficiently find video content related to a live television broadcast in real time using a query modeling approach?

To answer RQ6, we formalized the task of finding video content related to a live television broadcast as a Markov decision process and proposed a reinforcement learning approach to directly optimize retrieval effectiveness of queries generated from the stream of subtitles. We showed that our approach significantly improves retrieval effectiveness over state-of-the-art stationary baselines, while remaining sufficiently efficient to be used in near real time in a live television setting.

We showed that each component of the dynamic query modeling retrieval model contributes to the overall effectiveness. A larger number of weighted query terms significantly improve effectiveness. Static term features that are dependent on the target collection and background corpora are more important for selecting effective query terms than dynamic features derived from the stream of subtitles. Decaying term weights based on their recency further improves retrieval effectiveness.

Regarding our reinforcement learning approach we found that a medium explorative step size and a larger learning rate are the best choice in our setting. We have shown that optimizing nDCG also yields the best nDCG@5 scores. Lastly, we showed that our reinforcement learning based approach to dynamic query modeling still learns effectively when feedback becomes noisy. This suggests that our approach is also suited for use in a setting where feedback comes from user interaction.

8.2 Future Work

This thesis has resulted in insights and algorithms to ease the access to information. Beyond these, it opens up many interesting and important directions for future work. Below, we outline the main areas, along the three research themes.

Theme 1—Studying News Collections

Our initial research within this theme has been exploratory in nature. We motivated the research in Chapter 3 with exploratory search interfaces that provided support for researchers in studying and exploring large collections. Case studies of how these exploratory search interfaces were used served as motivation for research on connecting collections. In addition, we presented an algorithmic approach for finding frames in news.

Full evaluation of the exploratory search interfaces. We have left an analysis of the commonalities between the interfaces and a full evaluation of how such interfaces would be used by researchers for future work. We foresee several approaches for this evaluation, each with their own drawbacks and benefits. For example, comparing an exploratory search interface such as the one described in Chapter 3 to another interface that uses only a search box and a result list, is not the right approach, as these interfaces have a very different purpose. In the proposed contrastive evaluation, efficiency is not very important. In fact, users will probably not find things quicker, as the interface allows for casual browsing and exploring of a collection and is aimed at getting a deeper understanding of the collection. Evaluation should therefore focus on the quality of the insight that can be gained from using it. One approach for a detailed evaluation could be to study the separate components of the interface, to answer the questions: (1) Which components work, and which do not work? (2) Does interaction improve usability and effectiveness?

For evaluating the effectiveness of our approach, we can assess the quality of the insight gained from exploring the collection. For example, Bron et al. [35] compared the quality of research questions obtained using two variants of a exploratory search tool. For the task of document selection (described in Chapter 3), the quality of the selection of documents can be evaluated. We can measure the quality of this selection in terms of diversity, both temporal and non-temporal. One way of measuring this non-temporal diversity is to look at the framing of an issue, i.e., the ways in which issues are presented. A more diverse selection of documents on a issue will have a broader range of perspectives on it.

Supporting the exploration of other collections. Central to the motivational case study and algorithmic approach for connecting collections in Chapter 3 was a unique diachronic collection of digitized newspapers with proper document dating and rich metadata. Yet, there is nothing in our approaches that is collection-specific. We use two dimensions present in nearly every collection: time and textual content. We have released our work as open data and open-source software to foster future research, including on other collections. One can easily find examples in smaller timespans, e.g., investigating changing reputation of a company on Twitter, or finding when new words appear in a language, by analyzing books and news. The interesting question would be whether our approaches for connecting collections would work in such settings as well.

Extracting temporal references from historical narratives. We consider our proposed approach for using extracted temporal references to improve related article finding as just a first attempt. We have identified interesting challenges in extracting temporal references from historical narratives, such as the books of Loe de Jong. These books are written in a narrative style, where temporal references are often (partially) implicit. For example, a section on famine in the winter of 1944-1945 (referred to as the "hunger winter") often only refers only to days in these winter months, without referring to a year. Given the topic, a reader knows that January 15th refers to January 15th, 1945, but for an automatic approach, this is rather difficult to infer. In fact, half of the fourteen books indicate in the title that they cover only a specific period. Improving the accuracy of temporal reference extraction on such a collection poses interesting future work for information extraction researchers. How should you model and combine partial temporal references? What distance between references should you consider when combining references: a paragraph, a page, half a book? Is this perhaps different for running text than for section headers or the title?

Active learning to find frames in news. For the task of frame detection in news, we have shown that using an ensemble-based classification approach we are able to approach human performance in terms of accuracy on this task. A combined approach of human and automated frame detection seems to be the logical way forward. In such an active learning scenario, the selection of documents to annotate next is informed by the current classification model. An active learning approach could, for example, select the document for annotation that it is most uncertain about. An interesting research question would be whether using active learning would lead to reduced annotation effort for achieving the same accuracy. Furthermore, existing approaches for active learning typically consider a single classification task, whereas in the frame detection task we consider multiple frames per document. When selecting documents for annotation, would it be better to consider the current prediction for only a single frame or combine predictions for multiple frames and how would one combine this?

Uptake of tools. The tools presented in this research theme have been adopted in past and ongoing research projects. The BILAND project[1] aimed to make the tool from the case study in Chapter 3 useful for bilingual research on researching Eugenics in German and Dutch sources. The Translantis research project[2] aims to develop digital humanities approaches to study reference cultures. The program uses some of the tools developed for this thesis to analyze how the United States has served as a cultural model for the Netherlands in the twentieth century. A very interesting IR challenge in both projects is how to combine search results and aggregations from documents from two languages. The uptake and further development of the exploratory search tools developed within this research theme show that there is a clear need for such tools in the emerging field of digital humanities. New questions regarding how to critically assess the impact of such tools on historical research where raised and partially addressed during discussions at the Workshop on Tool Criticism in the Digital Humanities [211]. The exploratory search interface presented in Chapter 3 was considered as one of the use cases for this workshop. An avenue for future research would be to deeper understand how researchers from the humanities use such tools in their research methodology and what implications this has for IR system design.

Theme 2—Struggling and Success in Web Search

Our findings within this theme have implications for the design of search systems that help searchers struggle less and succeed more. Here, we discuss these implications and directions for future research that follow from this.

Direct application of reformulation strategies. We have demonstrated the capability to accurately predict the strategy associated with the next query reformulation (rather than syntactic transformations, as has traditionally been studied). This allows us to provide situation-specific search support at a higher (more strategic) level than syntactic query formulations. We left this application of our work for future research and foresee interesting questions regarding how to integrate such prediction in the search process and whether users will benefit from such support. For example, if we predict that a searcher is likely to perform an action such as adding an attribute, the system can focus on recommending queries with attributes in query suggestions or query auto-completions depending on when they are applied (and augmented with additional information about popularity and/or success if available from historic data). Would the relevancy of query suggestions improve and would searchers indeed struggling less?

Another direction for future research would be to leverage the automatic identification of pivotal queries for improving the search results or experience. Sets of (query → pivotal query) pairs can also be mined from log data. Such queries may also be leveraged internally within search engines (e.g., in blending scenarios, where the results from multiple queries are combined [178]) to help generate better quality search result lists, or present them as suggestions to searchers. Future work could answer whether users would click results from the pivotal query when blended in, whether pivotal queries are indeed good query suggestions, and whether either improves the user experience in web search.

[1] http://biland.science.uva.nl/

[2] http://translantis.wp.hum.uu.nl/

Hints and tips on reformulation strategies. As we demonstrated, struggling searchers, especially those destined to be unsuccessful, are highly likely to re-query. Learning the relationship between task success and the nature of the anticipated query reformulation allows search systems to generate human-readable hints about which types of reformulations to leverage (e.g., "add an action" leads to the highest proportion of pivotal queries, per Figure 5.14), and propose them in real time as people are searching. Moraveji et al. [153] showed that the presentation of optimal search hints, where they are presented for a task where they are known to have benefit, can have a lasting impact on searcher efficiency. Future research could investigate whether hints derived from the reformulation strategies are optimal and whether they have a lasting impact on searcher efficiency. These hints for reformulation strategies can be learned from all searchers, or perhaps even more interestingly, from advanced searchers [2, 226] or domain experts [228].

Other prediction tasks. Overall, accurate inferences and predictions about the nature of query reformulations can help searchers and search engines reduce struggling. Although our findings are promising, the nature and volume of our human-labeled data limited the types of the prediction tasks that we attempted. There are others, such as predicting search success given different query reformulation strategies that are interesting avenues for future work. Additional opportunities include working directly with searchers to better understand struggling in-situ, improving our classifiers, and experimenting with the integration of struggling support in search systems. This would give a broader insight into how users struggle and what support they could benefit from.

Theme 3—Pro-active Search for Live TV

To understand the broader applicability of our work within this theme, it helps to point out that the task of finding related content to a live television broadcast combines two traditional basic information retrieval tasks: search and filtering [16]. In a typical search task the query changes and the collection remains static. In a typical document filtering task, a standing query is used to filter a stream of documents. Our task concerns both. Similar tasks exist, such as summarizing social media in real time [182] and finding replications of news articles while they appear [212]. We believe that our linking and dynamic query modeling (DQM) approaches is applicable to those tasks too. A promising avenue for future work is to apply our algorithms and findings to other similar tasks, such as the two mentioned above. Furthermore, we identified extension to our approaches for future work. These are listed below.

Beyond talk shows and news. The datasets used in this theme consists of subtitles from video segments of talk shows and news broadcasts. We have chosen talk shows because they cover a broad range of topics, mostly current. However, our approach is not specific for talk shows and it will be interesting to evaluate this approach on different types of television broadcast, such as live events and sports. Furthermore, with advances in automatic speech recognition, combining our approach with the output of such a system may provide an effective solution when manually created subtitles are not available.

Enriching the context graph. We have shown that selecting which candidate links to add to the context graph is an important choice. An interesting follow-up to this observation is to further improve the representativeness of the context graph, for example by semantically enriching the graph, by weighting the edges of the graph, accounting for the quality of the evidence collected. We can further extend the enrichment to encode more information, e.g., by using the Wikipedia link structure. Adding edges in the context graph for links between Wikipedia pages could improve the coherence of the context graph. Future work could show whether the task of entity linking in streaming text benefits from such enrichments.

Learning from user interactions. A natural extension of our work within this theme is to explore learning from noisy feedback from actual user interactions. The DQM algorithm is designed with such feedback in mind. We have shown that we can deal with synthetic noisy feedback and still learn to generate effective queries when feedback becomes noisy. It would be interesting to see if this is also the case with actual noisy feedback from users. Developing an approach for entity linking in real time that can deal with noisy feedback from user interactions is an interesting challenge for future research. An open question is how to interpret the user feedback, e.g., is a click always a positive signal?

Generating multiple queries and retrieving diverse results. Our dynamic query modeling algorithm generates a single query at any point in time. Future work could address this by designing an extension that generates multiple queries and merges either queries or results. A contrastive experiment can show whether this is more effective. Similarly, an extension that explicitly generates temporally or topically diverse results may enhance the user experience. We could model this as a slot filling problem, where we have four video slots for which we select the most interesting videos to show to a user. Our evaluation considered this task a ranking task, but the slot filling problem fits closer with the actual user experience. The question is if such a view of the problem would lead to a different solution.

Uptake of algorithms. The framework for real-time entity linking developed within this research theme has been used by Dutch public broadcaster VARA for a Social TV application.[3] Here, links to background information where shown to television viewers who could then discuss around these topics with other viewers. A small scale evaluation suggested viewers enjoyed this experience and would use the application again. Future work on how users interact with such an application can provide interesting insights into this task and guide future research into entity linking in a streaming setting.

Semanticizer, the framework for entity linking that resulting from the research is described in more detail in the appendix and has been used by many researchers, e.g., to enrich search queries [81], books [136] and tweets [77, 221]. An interesting next step would be to investigate how much of the algorithmic approach is shared between these tasks and how much should be tailored for the specific task.

The approach for finding related archival video content for live television is used in a smart TV application for the NOS news.[4] The two novel applications for live television content show that our algorithms open up new opportunities and applications in this domain. Our algorithms represent a first step in supporting such applications. Future

[3] http://socialtv.vara.nl/

[4] http://weblogs.nos.nl/nieuwemedia/2014/05/27/nos-nieuws-app-voor-hbbtv-live/

work can build on insights gained from these applications and incorporate user feedback to design new algorithms and new application. Next steps would be to personalize recommended content and to better understand when a user would want to receive such recommendations.

APPENDIX

A Semanticizer

In this appendix, we describe the software developed for the research that was presented in Chapter 6. This software has been made available as an entity linking framework and webservice under the name *Semanticizer* on `http://semanticize.uva.nl`. Semanticizer includes the algorithms for entity linking using lexical matching on anchor text, detailed in Section 6.2. Below, we describe how to access the webservice (Section A.1), how to run the webservice yourself (Section A.2), and how to replicate a number of the experiments described in Chapter 6 (Section A.3).

A.1 Semanticizer Webservice API

This section describes how to use the Semanticizer webservice API on the basis of a number of example API calls. This REST-like webservice returns JSON and is publicly available at: `http://semanticize.uva.nl/api/`. We provide examples of HTTP calls, that you could, for example, use from the command line using cURL: `$ curl "http://semanticize.uva.nl/api/"`. To get started, we show how to perform entity linking on short text using the API. In this example, the text is send in an HTTP GET call. The text can also be sent as the body of a HTTP POST call. Below is the response of the Semanticizer and the requested URL, that specifies that we have a short Dutch text "UvA" and that we want the output to be 'pretty' (i.e. human readable).

```
            http://semanticize.uva.nl/api/nl?text=UvA&pretty
```

```
{
  "text": "UvA",
  "status": "OK",
  "links": [
    {
      "id": 14815,
      "text": "UvA",
      "label": "UvA",
      "title": "Universiteit van Amsterdam",
      "url": "http://nl.wikipedia.org/wiki/Universiteit van Amsterdam",
      "linkProbability": 0.2946058091286307,
      "priorProbability": 1.0,
      "senseProbability": 0.2946058091286307
    }
  ]
}
```

The Semanticizer webservice returns a JSON object containing the original text, a status message, and a list of links. The only link detected in this short text is a link to the Wikipedia page about the University of Amsterdam. Each link has a number of properties, including the substring of the submitted text based on which the link is made (‘text’ and ‘label’, see Normalization below), the title of the target Wikipedia page (‘title’) and a ‘url’ for that page. Finally, each link has three heuristic measures that estimate the likelihood of a link being correct for the ‘label’: ‘priorProbability’, ‘linkProbability’, and ‘senseProbability’. These three measures are discussed below and in detail in Section 6.2.1.

Counts. To illustrate how the three measures are computed we look at the raw numbers from which these are derived. For this, we add `counts` to the API call:

```
http://semanticize.uva.nl/api/nl?text=UvA&counts&pretty
```

```
{
  "text": "UvA",
  "status": "OK",
  "links": [
    {
      "id": 14815,
      "text": "UvA",
      "label": "UvA",
      "title": "Universiteit van Amsterdam",
      "url": "http://nl.wikipedia.org/wiki/Universiteit van Amsterdam",
      "linkProbability": 0.2946058091286307,
      "priorProbability": 1.0,
      "senseProbability": 0.2946058091286307,
      "docCount": 241,
      "occCount": 326,
      "linkDocCount": 71,
      "linkOccCount": 75,
      "senseDocCount": 71,
      "senseOccCount": 75
    }
  ]
}
```

We now have six new properties for the link:

occCount is the number of times the ‘label’ appears on Wikipedia.

docCount is the number of documents in which the ‘label’ appears on Wikipedia.

linkOccCount is the number of times the ‘label’ appears as anchor text for a link on Wikipedia.

linkDocCount is the number of documents in which the ‘label’ appears as anchor text for a link on Wikipedia.

senseOccCount is the number of times the ‘label’ appears as anchor text for a link to the Wikipedia page titled ‘title’.

senseDocCount is the number of documents in which the ‘label’ appears as anchor text for a link to the Wikipedia page titled ‘title’.

From these counts we compute the three heuristic measures that estimate the likelihood of a link being correct for the label. Note that we can have more than one link for the same label, each with a different title. The three heuristics are:

linkProbability is computed by dividing the 'linkDocCount' by 'docCount' and thus equals the proportion of documents in which the 'label' appears as anchor text for any link on Wikipedia (Equation 6.1).

priorProbability is computed by dividing 'senseDocCount' by 'linkDocCount' and thus equals the proportion of documents where the Wikipedia page titled 'title' is the target when 'label' appears as anchor text on Wikipedia (Equation 6.2).

senseProbability is the product of 'linkProbability' and 'priorProbability' and thus equals the proportion of documents where 'label' is used as anchor text to link to the Wikipedia page titled 'title' out of all pages where 'label' appears on Wikipedia (Equation 6.3).

Conceptually, you can regard 'linkProbability' as a measure of how likely 'label' is to be a link, 'priorProbability' as a measure of how unambiguous 'label' is, and 'senseProbability' as a measure combining both characteristics. Adding `lowerConfidenceBound` to the API call will change these measures to their lower bound estimate, based on a Wilson 95% confidence interval [229] (described in Section 6.2.1 and Equation 6.4). The experiments presented in Table 6.5 show that this improves effectiveness for nearly all measures.

`Normalize.` The input string is normalized before entity linking is performed. The same normalization is applied on the input string as on the anchor text in Wikipedia. Three complementary normalization methods are provided:

dash will replace all dashes (-) with a space, making 'Déjà-vu-feeling' the same as 'Déjà vu feeling'.

accents will remove accents from characters and convert the string to normal form KD (NFKD).[1] This will yield in 'Déjà vu' being equal to 'Deja vu' after normalization.

lower will lowercase all text, making 'Déjà vu' the same as 'déjà vu'.

By default 'dash' and 'accents' are applied, but different combinations can be used, e.g., via `http://semanticize.uva.nl/api/nl?text=UvA&normalize=lower,dash,accents`.

`Filter.` The link retrieval model for real-time entity linking implemented in the Semanticizer and described in Section 6.2.1 is recall-oriented. This means that we produce many link candidates, even if they are not very relevant. In fact, if a label is used as anchor text on Wikipedia just once and occurs in the input text, it is considered a link candidate. Clearly, it is important to rank and filter the link candidates that are produced. For these the heuristic measures mentioned above can be used. Alternatively, adding `largestMatching` to the API call will return only the links with the largest matching anchor text and ignore all constituent anchors, commonly known as the CMNS approach [149]. Our experiments on pruning, described in Section 6.4.1, show that a threshold on 'senseProbability' is best suited for pruning. We can also ensure that a label

[1] `http://www.unicode.org/reports/tr44/tr44-4.html`

is used as a link in at least a minimal number of documents (say 5 times). You can apply a filter for this in the API call:

```
       http://semanticize.uva.nl/api/nl?text=Karel de Grote&pretty
              &filter=senseProbability>0.3,linkDocCount>=5
```

```
{
  "text": "Karel de Grote",
  "status": "OK",
  "links": [
    {
      "id": 5337,
      "text": "Karel de Grote",
      "label": "Karel de Grote",
      "title": "Karel de Grote",
      "url": "http://nl.wikipedia.org/wiki/Karel de Grote",
      "linkProbability": 0.8417582417582418,
      "priorProbability": 0.9989094874591058,
      "senseProbability": 0.8417582417582418,
    }
  ]
}
```

`Context.` Chapter 6 presented approaches for explicitly modeling context in a streaming textual setting. For this, the Semanticizer allows you to specify a context with your API call. This influences the computation of features (described below) and allows you to filter links to make sure they are unique in their context:

```
   http://semanticize.uva.nl/api/nl?context=test&filter=unique&text=UvA
```

```
{
  "text": "UvA",
  "status": "OK",
  "links": [
    {
      "id": 14815,
      "text": "UvA",
      "label": "UvA",
      "title": "Universiteit van Amsterdam",
      "url": "http://nl.wikipedia.org/wiki/Universiteit van Amsterdam",
      "linkProbability": 0.2946058091286307,
      "priorProbability": 1.0,
      "senseProbability": 0.2946058091286307
    }
  ]
}
```

Doing the same request again will result in no links:

```
   http://semanticize.uva.nl/api/nl?context=test&filter=unique&text=UvA
```

```
{
  "text": "UvA",
  "status": "OK",
  "links": []
}
```

Features. Adding `features` to your API call will make Semanticizer compute and return features for each link. See Tables 6.1 and 6.2 for an overview of the link features and contextual features respectively.

A.2 Running the Webservice

The Semanticizer source code[2] is released under LGPL license. If you want to dive into the code, start at `semanticizer/server/__main__.py`. Semanticizer has been tested with Python 2.7.3 on Mac OS X 10.8 and on Linux (RedHat EL5, Debian jessie/sid and Ubuntu 12.04). To start the webservice, follow these five steps:

1. The following Python modules need to be installed (using easy_install or pip): nltk, leven, networkx, lxml, flask, redis (optional, see step 3), scikit-learn (optional, see step 5), scipy (optional, see step 5), mock (optional, used by the tests).
2. A summary of a Wikipedia dump is needed. To obtain this, download the Wikipedia Miner [149] CSV files from `http://sourceforge.net/projects/wikipedia-miner/files/data/`.
3. Copy one of the two config files in the conf folder to `semanticizer.yml` in that folder and adapt to your situation. You have the choice of loading all data into memory (use `semanticizer.memory.yml`) or into Redis using the following steps:

   (a) Copy `semanticizer.redis.yml` into `semanticizer.yml`.

   (b) Set up and start a Redis server.

   (c) Load data into Redis: `python -m semanticizer.dbinsert`.
4. Run the server using `python -m semanticizer.server`.
5. In order to work with the experimental functionality for machine learning, you need to install the scikit-learn and scipy packages.

A.3 Replicating Experiments

In this section, we describe how to replicate the Experiments 1–3, described in Section 6.3. In these experiment, we evaluate link retrieval using lexical matching on anchor text on television subtitles. We first describe how to obtain the dataset and then explain how to map the experiments to Semanticizer API calls.

Dataset. The dataset that was described in Section 6.3 has been made available to the research community;[3] it consists of more than 1,500 manually annotated links in over 5,000 subtitle chunks for 50 video segments from six episodes of a live daily talk show. Each video segment has a unique ID and three associated files (the first two are provided, the last one needs to be reconstructed):

[2] For code, see: `https://github.com/semanticize/semanticizer/`.

[3] See: `http://ilps.science.uva.nl/resource/oair-2013-feeding-second-screen`.

`<UniqueID>.source.txt` describes when the episode that contains the video segment was originally broadcasted and provides a URL to the archived broadcast. It also describes the start and end time of the segment in the original broadcast and the number of chunks.

`<UniqueID>.positives.txt` lists the titles of all target Wikipedia pages that have been manually annotated as relevant to a broad viewer audience, as described in Section 6.3.

`<UniqueID>.txt` contains a subtitle chunk per line. Due to copyright restrictions, this file was not released with the dataset, but it is reproducible from the archived broadcast.

Running experiments. To replicate the experiments 1–3 from Chapter 6, you need the dataset described above. Next you can decide on the Semanticizer API parameters that correspond to the variant of the approach that you want to replicate. Examples of these parameters are listed in Table A.1). With the dataset and parameters, you can replicate the results by taking the following steps:

1. For each video segment, first obtain all the link candidates:
   (a) Perform a Semanticizer API call for each subtitle chunk, with the chunk as text and the parameters for the specific variant. For example:
   `/api/nl?text=Goedenavond, Dames en Heren&largestMatching`.
   (b) Combine the results for each chunk, keeping only the 'title' of each link candidate and a measure for ranking (e.g., 'senseProbability').
   (c) Compute metrics by comparing the ranked results against the ground truth in `<UniqueID>.positives.txt`.
2. Average the metrics across the video segments.

This same approach can be used to replicate the experiments on reranking link candidates. The API call in step 1a should then include `features`, and the ranking in step 1b should be replaced with a learning to rank model. For the experiments in Chapter 6, we used five-fold cross-validation at the video segment level (See Section 6.3).

Table A.1: Examples of API calls that map to specific variants of the recall-oriented link candidate finding substep, evaluated in experiments 1 and 2 as described in Section 6.3 and presented in Table 6.4 on the specified line number.

Variant	Semanticizer API call
4. Lexical match on normalized anchor	`/api/nl`
5. Largest matching n-gram only	`/api/nl?largestMatching`
7. Threshold on *SENSEPROB*	`/api/nl?filter=senseProbability>=0.002`

Bibliography

[1] E. Agapie, G. Golovchinsky, and P. Qvarfordt. Leading people to longer queries. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 3019–3022, New York, NY, USA, 2013. ACM. (Cited on pages 27 and 72.)

[2] M. Ageev, Q. Guo, D. Lagun, and E. Agichtein. Find it if you can: A game for modeling different types of web search success using interaction data. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 345–354, New York, NY, USA, 2011. ACM. (Cited on pages 25, 26, 27, 68, 69, 88, and 142.)

[3] W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski. Visualizing time-oriented data—a systematic view. *Computers & Graphics*, 31(3):401 – 409, 2007. (Cited on page 22.)

[4] J. Allan. *Automatic hypertext construction*. PhD thesis, Cornell University, 1995. (Cited on page 29.)

[5] J. Allan. *Topic Detection and Tracking: Event-Based Information Organization*. Kluwer international series on information retrieval. Springer US, 2002. (Cited on page 19.)

[6] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study final report. Computer Science Department Paper 341, CMU, 1998. (Cited on pages 28 and 116.)

[7] O. Alonso, J. Strötgen, R. Baeza-Yates, and M. Gertz. Temporal information retrieval: Challenges and opportunities. In *Proceedings of the 1st International Temporal Web Analytics Workshop (TWAW) at WWW'11*, pages 1–8, 2011. (Cited on pages 17 and 23.)

[8] P. Anick. Using terminological feedback for web search refinement: A log-based study. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 88–95, New York, NY, USA, 2003. ACM. (Cited on pages 27 and 69.)

[9] R. Artstein and M. Poesio. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596, 2008. (Cited on page 80.)

[10] C.-m. Au Yeung and A. Jatowt. Studying how the past is remembered: Towards computational history through large scale text mining. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 1231–1240, New York, NY, USA, 2011. ACM. (Cited on pages 19 and 20.)

[11] A. Aula, R. M. Khan, and Z. Guan. How does search behavior change as search becomes more difficult? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 35–44, New York, NY, USA, 2010. ACM. (Cited on pages 4, 25, 26, 67, 68, 69, and 75.)

[12] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval: The Concepts and Technology Behind Search*. Addison Wesley, 2011. (Cited on pages 1, 2, 13, 16, 18, and 19.)

[13] N. Balasubramanian and S. Cucerzan. Topic pages: An alternative to the ten blue links. In *ICSC '10*, 2010. (Cited on page 29.)

[14] N. Balasubramanian, G. Kumaran, and V. R. Carvalho. Exploring reductions for long web queries. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 571–578, New York, NY, USA, 2010. ACM. (Cited on page 31.)

[15] A. G. Barto, R. S. Sutton, and P. S. Brouwer. Associative search network: A reinforcement learning associative memory. *Biological Cybernetics*, 40(3):201–211, 1981. (Cited on pages 16 and 117.)

[16] N. J. Belkin and W. B. Croft. Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM*, 35(12):29–38, Dec. 1992. (Cited on pages 17, 133, and 142.)

[17] R. E. Bellman. A markovian decision process. *Journal of Applied Mathematics and Mechanics*, 6: 679–684, 1957. (Cited on pages 16, 115, and 117.)

[18] M. Bendersky and W. B. Croft. Discovering key concepts in verbose queries. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 491–498, New York, NY, USA, 2008. ACM. (Cited on pages 31 and 116.)

[19] M. Bendersky, D. Metzler, and W. B. Croft. Parameterized concept weighting in verbose queries. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 605–614, New York, NY, USA, 2011. ACM. (Cited on pages 31 and 116.)

[20] M. Bendersky, L. Garcia-Pueyo, J. Harmsen, V. Josifovski, and D. Lepikhin. Up next: Retrieval methods for large scale related video suggestion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1769–1778, New York, NY, USA, 2014. ACM. (Cited on pages 31 and 116.)

[21] P. N. Bennett, K. Svore, and S. T. Dumais. Classification-enhanced ranking. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 111–120, New York, NY, USA, 2010. ACM. (Cited on page 72.)

[22] P. N. Bennett, F. Radlinski, R. W. White, and E. Yilmaz. Inferring and using location metadata to personalize web search. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 135–144, New York, NY, USA, 2011. ACM. (Cited on page 2.)

[23] R. Berendsen, E. Meij, D. Odijk, M. de Rijke, and W. Weerkamp. The University of Amsterdam at TREC 2012. In *Proceedings of the 21st Text REtrieval Conference (TREC 2012)*, 2012. (Cited on page 12.)

[24] D. C. Blair. Searching biases in large interactive document retrieval systems. *Journal of the American Society for Information Science*, 31(4):271–277, 1980. (Cited on page 76.)

[25] R. Blanco, G. D. F. Morales, and F. Silvestri. Intonews: Online news retrieval using closed captions. *Information Processing & Management*, 51(1):148–162, 2015. (Cited on pages 28, 29, 30, 43, 93, 99, 100, 113, 115, 116, 124, and 125.)

[26] R. Blanco, G. Ottaviano, and E. Meij. Fast and space-efficient entity linking for queries. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 179–188, New York, NY, USA, 2015. ACM. (Cited on pages 90, 100, and 104.)

[27] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna. The query-flow graph: Model and applications. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 609–618, New York, NY, USA, 2008. ACM. (Cited on page 30.)

[28] T. Brants and A. Franz. Web 1T 5-gram 10 european languages version 1 LDC2009T25. Philadelphia: Linguistic Data Consortium, 2009. (Cited on page 119.)

[29] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. (Cited on page 95.)

[30] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and regression trees*. Wadsworth, 1984. (Cited on page 85.)

[31] C. Brew and S. Schulte im Walde. Spectral clustering for german verbs. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 117–124, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. (Cited on page 30.)

[32] P. R. Brewer. Framing, value words, and citizens' explanations of their issue opinions. *Political Communication*, 19(3):303–316, 2002. (Cited on pages 25 and 55.)

[33] M. Bron, B. Huurnink, and M. de Rijke. Linking archives using document enrichment and term selection. In S. Gradmann, F. Borri, C. Meghini, and H. Schuldt, editors, *Research and Advanced Technology for Digital Libraries: International Conference on Theory and Practice of Digital Libraries*, TPDL 2011, pages 360–371, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. (Cited on pages 23, 29, 31, 43, 90, and 116.)

[34] M. Bron, J. van Gorp, F. Nack, and M. de Rijke. Exploratory search in an audio-visual archive. In *EuroHCIR2013: the 3rd European workshop on human-computer interaction and information retrieval*, 2011. (Cited on page 22.)

[35] M. Bron, J. van Gorp, F. Nack, M. de Rijke, A. Vishneuski, and S. de Leeuw. A subjunctive exploratory search interface to support media studies researchers. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 425–434, New York, NY, USA, 2012. ACM. (Cited on pages 22, 23, and 139.)

[36] E. Brown, S. Srinivasan, A. Coden, D. Ponceleon, J. Cooper, A. Amir, and J. Pieper. Towards speech as a knowledge resource. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, CIKM '01, pages 526–528, New York, NY, USA, 2001. ACM. (Cited on page 28.)

[37] R. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)*, pages 9–16. ACL, 2006. (Cited on page 29.)

[38] B. Burscher, D. Odijk, R. Vliegenthart, M. de Rijke, and C. H. de Vreese. Teaching the computer to code frames in news: Comparing two supervised machine learning approaches to frame analysis. *Communication Methods and Measures*, 8(3):190–206, 2014. (Cited on page 11.)

[39] V. Bush. As we may think. *Atlantic Monthly*, 1945. (Cited on page 14.)

[40] S. Büttcher, C. Clarke, and G. V. Cormack. *Information Retrieval: Implementing and Evaluating Search Engines*. The MIT Press, 2010. (Cited on pages 18 and 19.)

[41] L. Byron and M. Wattenberg. Stacked graphs–geometry & aesthetics. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1245–1252, 2008. (Cited on page 46.)

[42] J. Callan et al. The clueweb12 dataset. `http://lemurproject.org/clueweb12/`, 2012. (Cited on page 18.)

[43] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999. (Cited on page 47.)

[44] B. Carterette. System effectiveness, user models, and user utility: A conceptual framework for investigation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 903–912, New York, NY, USA, 2011. ACM. (Cited on pages 18 and 19.)

[45] B. Carterette and R. Jones. Evaluating search engines by modeling the relationship between relevance and clicks. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 217–224. Curran Associates, Inc., 2008. (Cited on page 124.)

[46] D. O. Case. *Looking for information: A survey of research on information seeking, needs and behavior*. Emerald Group Publishing, 2012. (Cited on page 14.)

[47] A.-S. Cheng, K. R. Fleischmann, P. Wang, and D. W. Oard. Advancing social science research by applying computational linguistics. In *Proceedings of the Annual Conference of the American Society for Information Science and Technology*, 2008. (Cited on pages 23 and 53.)

[48] A. Chuklin, I. Markov, and M. de Rijke. *Click Models for Web Search*. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers, August 2015. (Cited on page 18.)

[49] K. W. Church and W. A. Gale. Poisson mixtures. *Natural Language Engineering*, 1:163–190, 1995. (Cited on page 119.)

[50] C. W. Cleverdon. Report on the testing and analysis of an investigation into the comparative efficiency of indexing systems. In *ASLIB Proceedings*, 1962. (Cited on pages 17 and 18.)

[51] C. Condit. *The meanings of the gene: Public debates about human heredity*. University of Wisconsin Press, 1999. (Cited on pages 21, 35, and 36.)

[52] P. Courant, S. Fraser, M. Goodchild, et al. Our cultural commonwealth. Techn. report, American Council of Learned Societies, 2006. (Cited on pages 21 and 35.)

[53] N. Craswell and M. Szummer. Random walks on the click graph. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 239–246, New York, NY, USA, 2007. ACM. (Cited on page 30.)

[54] W. B. Croft, D. Metzler, and T. Strohman. *Search engines: Information retrieval in practice*. Addison-Wesley Reading, 2010. (Cited on pages 13, 14, 17, 18, and 19.)

[55] S. Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP '07, pages 708–716. ACL, 2007. (Cited on pages 29 and 30.)

[56] V. de Boer, J. van Doornik, L. Buitinck, M. Marx, T. Veken, and K. Ribbens. Linking the kingdom: Enriched access to a historiographical text. In *Proceedings of the Seventh International Conference on Knowledge Capture*, K-CAP '13, pages 17–24, New York, NY, USA, 2013. ACM. (Cited on page 40.)

[57] O. de Rooij, D. Odijk, and M. de Rijke. Themestreams: Visualizing the stream of themes discussed in politics. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 1077–1078, New York, NY, USA, 2013. ACM. (Cited on pages 11, 12, and 51.)

[58] D. Downey, S. Dumais, D. Liebling, and E. Horvitz. Understanding the relationship between searchers' queries and information goals. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 449–458, New York, NY, USA, 2008. ACM. (Cited on pages 15, 26, and 70.)

[59] W. M. Duff and C. A. Johnson. Accidentally found on purpose: Information-seeking behavior of historians in archives. *The Library Quarterly: Information, Community, Policy*, 72(4):472–496, 2002. (Cited on pages 3, 21, 35, 38, 39, and 136.)

[60] S. Dumais. Task-based search: a search engine perspective. NSF Workshop on Task-Based Search, March 2013. (Cited on pages 1 and 2.)

[61] M. Efron. Linear time series models for term weighting in information retrieval. *Journal of the American Society for Information Science and Technology*, 61(7):1299–1312, 2010. (Cited on page 17.)

[62] C. Eickhoff, J. Teevan, R. White, and S. Dumais. Lessons from the journey: A query log analysis of within-session learning. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 223–232, New York, NY, USA, 2014. ACM. (Cited on pages 26 and 69.)

[63] G. Erkan and D. Radev. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479, 2004. (Cited on page 30.)

[64] H. Feild and J. Allan. Task-aware query recommendation. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 83–92, New York, NY, USA, 2013. ACM. (Cited on pages 2 and 27.)

[65] H. A. Feild, J. Allan, and R. Jones. Predicting searcher frustration. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR

'10, pages 34–41, New York, NY, USA, 2010. ACM. (Cited on pages 25, 26, 68, and 69.)

[66] E. Feinberg and A. Shwartz. *Handbook of Markov Decision Processes*. Kluwer, 2002. (Cited on pages 16 and 117.)

[67] P. Ferragina and U. Scaiella. Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1625–1628, New York, NY, USA, 2010. ACM. (Cited on pages 30, 89, and 102.)

[68] J. Firth. A synopsis of linguistic theory 1930-1955. In *Studies in Linguistic Analysis*, pages 1–32. Oxford: Philological Society, 1957. (Cited on page 20.)

[69] S. Fissaha Adafre and M. de Rijke. Discovering missing links in wikipedia. In *Proceedings of the 3rd International Workshop on Link Discovery*, LinkKDD '05, pages 90–97, New York, NY, USA, 2005. ACM. (Cited on page 29.)

[70] K. R. Fleischmann, D. W. Oard, A.-S. Cheng, P. Wang, and E. Ishita. Automatic classification of human values: Applying computational thinking to information ethics. *Proceedings of the American Society for Information Science and Technology*, 46(1):1–4, 2009. (Cited on pages 20 and 53.)

[71] E. A. Fox. Characterization of two new experimental collections in computer and information science containing textual and bibliographic concepts. Techn. report, Cornell University, 1983. (Cited on page 17.)

[72] S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM Transactions on Information Systems*, 23(2):147–168, Apr. 2005. (Cited on pages 25, 26, 68, 71, 87, and 88.)

[73] N. Fuhr. Probabilistic models in information retrieval. *The Computer Journal*, 35(3):243–255, 1992. (Cited on page 16.)

[74] J. Galtung and M. H. Ruge. The structure of foreign news. *Journal of Peace Research*, 2(1):64–90, 1965. (Cited on pages 24 and 54.)

[75] W. Gamson. *Talking Politics*. Cambridge University Press, 1992. (Cited on pages 25 and 54.)

[76] W. A. Gamson and A. Modigliani. Media discourse and public opinion on nuclear power: A constructionist approach. *American Journal of Sociology*, 95(1):1–37, 1989. (Cited on pages 4, 24, 53, 54, and 137.)

[77] C. Gârbacea, D. Odijk, D. Graus, I. Sijaranamual, and M. de Rijke. Combining multiple signals for semanticizing tweets: University of Amsterdam at #Microposts2015. In *5th workshop on 'Making Sense of Microposts' at World Wide Web Conference 2015*, 2015. (Cited on pages 12 and 143.)

[78] G. J. Garraghan. *A Guide to Historical Method*. Fordham University Press: New York, 1946. (Cited on pages 3, 21, 35, 39, and 136.)

[79] Google. The new multi-screen world: Understanding cross-platform consumer behaviour, August 2012. `http://services.google.com/fh/files/misc/multiscreenworld_final.pdf` [Online; accessed March 2013]. (Cited on pages 5 and 27.)

[80] D. Graus, M.-H. Peetz, D. Odijk, O. de Rooij, and M. de Rijke. yourhistory–semantic linking for a personalized timeline of historic events. In *Proceedings of the LinkedUp Veni Competition on Linked and Open Data for Education held at the Open Knowledge Conference (OKCon 2013)*, CEUR-WS, 2013. (Cited on page 12.)

[81] D. Graus, D. Odijk, M. Tsagkias, W. Weerkamp, and M. de Rijke. Semanticizing search engine queries: The university of amsterdam at the erd 2014 challenge. In *Proceedings of the First International Workshop on Entity Recognition & Disambiguation*, ERD '14, pages 69–74, New York, NY, USA, 2014. ACM. (Cited on pages 12 and 143.)

[82] S. Green. Building hypertext links by computing semantic similarity. *IEEE Transactions on Knowledge and Data Engineering*, 11(5):713–730, 1999. (Cited on page 29.)

[83] D. Guan, S. Zhang, and H. Yang. Utilizing query change for session search. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 453–462, New York, NY, USA, 2013. ACM. (Cited on pages 16 and 117.)

[84] J. Guo, G. Xu, H. Li, and X. Cheng. A unified and discriminative model for query refinement. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 379–386, New York, NY, USA, 2008. ACM. (Cited on pages 27 and 69.)

[85] Q. Guo, R. W. White, Y. Zhang, B. Anderson, and S. T. Dumais. Why searchers switch: Understanding and predicting engine switching rationales. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 335–344, New York, NY, USA, 2011. ACM. (Cited on page 27.)

[86] M. Gupta and M. Bendersky. Information retrieval with verbose queries. *Foundations and Trends in*

Information Retrieval, 9(3-4):209–354, 2015. (Cited on page 31.)

[87] F. Hasibi, K. Balog, and S. E. Bratsberg. Entity linking in queries: Tasks and evaluation. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, ICTIR '15, pages 171–180, New York, NY, USA, 2015. ACM. (Cited on pages 30, 90, 100, and 104.)

[88] A. Hassan. Identifying Web Search Query Reformulation using Concept based Matching. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, EMNLP '13, pages 1000–1010, 2013. (Cited on page 75.)

[89] A. Hassan, R. Jones, and K. L. Klinkner. Beyond DCG: User behavior as a predictor of a successful search. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 221–230, New York, NY, USA, 2010. ACM. (Cited on pages 25, 26, 68, 69, and 71.)

[90] A. Hassan, X. Shi, N. Craswell, and B. Ramsey. Beyond clicks: Query reformulation as a predictor of search satisfaction. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 2019–2028, New York, NY, USA, 2013. ACM. (Cited on pages 26, 27, and 69.)

[91] A. Hassan, R. W. White, and Y.-M. Wang. Toward self-correcting search engines: Using underperforming queries to improve search. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 263–272, New York, NY, USA, 2013. ACM. (Cited on pages 26 and 27.)

[92] A. Hassan, R. W. White, S. T. Dumais, and Y.-M. Wang. Struggling or exploring?: Disambiguating long search sessions. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 53–62, New York, NY, USA, 2014. ACM. (Cited on pages 1, 2, 4, 25, 26, 67, 69, 70, 72, and 88.)

[93] S. Havre, B. Hetzler, and L. Nowell. Themeriver: Visualizing theme changes over time. In *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on*, pages 115–123. IEEE, 2000. (Cited on page 19.)

[94] J. He, M. de Rijke, M. Sevenster, R. van Ommering, and Y. Qian. Generating links to background knowledge: A case study using narrative radiology reports. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 1867–1876, New York, NY, USA, 2011. ACM. (Cited on pages 29, 30, and 90.)

[95] M. A. Hearst. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64, March 1997. (Cited on pages 28, 99, and 116.)

[96] M. Henzinger, B.-W. Chang, B. Milch, and S. Brin. Query-free news search. *World Wide Web*, 8(2): 101–126, June 2005. (Cited on pages 19, 28, 29, 43, 115, and 116.)

[97] K. Hofmann, S. Whiteson, and M. de Rijke. Balancing exploration and exploitation in listwise and pairwise online learning to rank for information retrieval. *Information Retrieval*, 16(1):63–90, 2013. (Cited on pages 16 and 117.)

[98] J. Huang and E. N. Efthimiadis. Analyzing and evaluating query reformulation strategies in web search logs. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 77–86, New York, NY, USA, 2009. ACM. (Cited on pages 27, 69, 75, 83, and 84.)

[99] S. B. Huffman and M. Hochster. How well does result relevance predict session satisfaction? In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 567–574, New York, NY, USA, 2007. ACM. (Cited on page 26.)

[100] B. Huurnink, L. Hollink, W. van den Heuvel, and M. de Rijke. Search behavior of media professionals at an audiovisual archive: A transaction log analysis. *Journal of the American Society for Information Science and Technology*, 61(6):1180–1197, June 2010. (Cited on page 29.)

[101] P. Ingwersen and K. Järvelin. *The Turn: Integration of Information Seeking and Retrieval in Context*. Springer Netherlands, Dordrecht, 2005. (Cited on pages 1, 2, 13, 14, 15, 17, and 18.)

[102] S. Iyengar. *Is anyone responsible?: How television frames political issues*. University of Chicago Press, 1994. (Cited on pages 24 and 54.)

[103] B. J. Jansen. The methodology of search log analysis. *Handbook of research on Web log analysis*, pages 100–123, 2008. (Cited on page 15.)

[104] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20(4):422–446, Oct. 2002. (Cited on pages 18, 19, and 124.)

[105] J. Jiang, A. Hassan Awadallah, X. Shi, and R. W. White. Understanding and predicting graded search satisfaction. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 57–66, New York, NY, USA, 2015. ACM. (Cited on pages 25, 26, and 69.)

[106] V. Jijkoun, M. A. Khalid, M. Marx, and M. de Rijke. Named entity normalization in user generated content. In *Proceedings of the Second Workshop on Analytics for Noisy Unstructured Text Data*, AND '08, pages 23–30, New York, NY, USA, 2008. ACM. (Cited on page 30.)

[107] V. Jijkoun, M. de Rijke, W. Weerkamp, P. Ackermans, and G. Geleijnse. Mining user experiences from online forums: an exploration. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*, pages 17–18. Association for Computational Linguistics, 2010. (Cited on pages 20 and 53.)

[108] X. Jin, M. Sloan, and J. Wang. Interactive exploratory search for multi page search results. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 655–666, New York, NY, USA, 2013. ACM. (Cited on page 16.)

[109] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 133–142, New York, NY, USA, 2002. ACM. (Cited on page 56.)

[110] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 154–161, New York, NY, USA, 2005. ACM. (Cited on page 18.)

[111] R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, pages 387–396, New York, NY, USA, 2006. ACM. (Cited on pages 27 and 88.)

[112] D. Kelly. Methods for evaluating interactive information retrieval systems with users. *Foundations and Trends in Information Retrieval*, 3(1–2):1–224, 2009. (Cited on pages 18 and 26.)

[113] T. Kenter, M. Wevers, P. Huijnen, and M. de Rijke. Ad hoc monitoring of vocabulary shifts over time. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 1191–1200, New York, NY, USA, 2015. ACM. (Cited on page 20.)

[114] Y. Kim, A. Hassan, R. W. White, and I. Zitouni. Modeling dwell time to predict click-level satisfaction. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 193–202, New York, NY, USA, 2014. ACM. (Cited on pages 26, 70, and 87.)

[115] Y. Kim, J. Seo, W. B. Croft, and D. A. Smith. Automatic suggestion of phrasal-concept queries for literature search. *Information Processing & Management*, 50(4):568–583, 2014. (Cited on pages 31, 43, 116, and 132.)

[116] M. Kleppe, L. Hollink, M. Kemman, D. Juric, H. Beunders, J. Blom, J. Oomen, and G. J. Houben. Polimedia. analysing media coverage of political debates by automatically generated links to radio & newspaper items. In *Proceedings of the LinkedUp Veni Competition, on Linked and Open Data for Education*. CEUR-WS, 2014. (Cited on pages 23 and 43.)

[117] W. Krauth and M. Mézard. Learning algorithms with optimal stability in neural networks. *Journal of Physics A: Mathematical and General*, 20(11):L745, 1999. (Cited on page 56.)

[118] G. Kumaran and V. R. Carvalho. Reducing long queries using query quality predictors. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 564–571, New York, NY, USA, 2009. ACM. (Cited on pages 31 and 116.)

[119] D. Laniado and P. Mika. Making sense of twitter. In P. F. Patel-Schneider, Y. Pan, P. Hitzler, P. Mika, L. Zhang, J. Z. Pan, I. Horrocks, and B. Glimm, editors, *9th International Semantic Web Conference*, ISWC 2010, pages 470–485, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. (Cited on page 29.)

[120] M. Larson, E. Newman, and G. J. F. Jones. Overview of VideoCLEF 2009: New perspectives on speech-based multimedia content enrichment. In C. Peters, B. Caputo, J. Gonzalo, G. J. F. Jones, J. Kalpathy-Cramer, H. Müller, and T. Tsikrika, editors, *Multilingual Information Access Evaluation II. Multimedia Experiments: 10th Workshop of the Cross-Language Evaluation Forum*, CLEF 2009, pages 354–368, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. (Cited on pages 30 and 95.)

[121] H. D. Lasswell. The structure and function of communication in society. *The Communication of Ideas*, 37, 1948. (Cited on pages 4, 23, 53, and 137.)

[122] T. Lau and E. Horvitz. Patterns of search: Analyzing and modeling web query refinement. In *Proceedings of the Seventh International Conference on User Modeling*, UM '99, pages 119–128, Secaucus, NJ, USA, 1999. Springer-Verlag New York, Inc. (Cited on pages 27, 69, and 75.)

[123] D. Lazer, A. S. Pentland, L. Adamic, S. Aral, A. L. Barabasi, D. Brewer, N. Christakis, N. Contractor, J. Fowler, M. Gutmann, et al. Life in the network: the coming age of computational social science. *Science*, 323(5915):721, 2009. (Cited on pages 20, 23, and 53.)

[124] M. Lease. An improved markov random field model for supporting verbose queries. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 476–483, New York, NY, USA, 2009. ACM. (Cited on pages 31 and 116.)

[125] M. Lease, J. Allan, and W. B. Croft. Regression rank: Learning to meet the opportunity of descriptive queries. In M. Boughanem, C. Berrut, J. Mothe, and C. Soule-Dupuy, editors, *Advances in Information*

Retrieval: 31th European Conference on IR Research, ECIR 2009, pages 90–101, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. (Cited on pages 31, 115, 116, 117, 118, 124, and 125.)

[126] C.-J. Lee and W. B. Croft. Generating queries from user-selected text. In *Proceedings of the 4th Information Interaction in Context Symposium*, IIIX '12, pages 100–109, New York, NY, USA, 2012. ACM. (Cited on pages 31 and 116.)

[127] J. Lensen. De zoektocht naar het midden: nieuwe perspectieven op de herinnering aan de Tweede Wereldoorlog in Vlaanderen en Duitsland. *Internationale Neerlandistiek*, 52(2):113–133, 2014. (Cited on page 21.)

[128] Z. Liao, Y. Song, L.-w. He, and Y. Huang. Evaluating the effectiveness of search task trails. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 489–498, New York, NY, USA, 2012. ACM. (Cited on pages 2 and 27.)

[129] Q. Liu, E. Agichtein, G. Dror, Y. Maarek, and I. Szpektor. When web search fails, searchers become askers: Understanding the transition. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 801–810, New York, NY, USA, 2012. ACM. (Cited on page 27.)

[130] T.-Y. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3 (3):225–331, 2009. (Cited on page 16.)

[131] S. Lopez and C. Snyder. *The Oxford Handbook of Positive Psychology*. Oxford University Press, 2011. (Cited on page 25.)

[132] J. Luo, S. Zhang, and H. Yang. Win-win search: Dual-agent stochastic game in session search. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, pages 587–596, New York, NY, USA, 2014. ACM. (Cited on pages 16 and 117.)

[133] J. Luo, S. Zhang, X. Dong, and H. Yang. Designing states, actions, and rewards for using pomdp in session search. In A. Hanbury, G. Kazai, A. Rauber, and N. Fuhr, editors, *Advances in Information Retrieval: 37th European Conference on IR Research*, ECIR 2015, pages 526–537, Cham, 2015. Springer International Publishing. (Cited on pages 16 and 117.)

[134] C. D. Manning, P. Raghavan, H. Schütze, et al. *Introduction to information retrieval*. Cambridge University Press, 2008. (Cited on pages 1, 2, 13, 18, and 19.)

[135] G. Marchionini. Exploratory search: From finding to understanding. *Communications of the ACM*, 49(4): 41–46, Apr. 2006. (Cited on pages 2, 3, 22, 25, 52, and 67.)

[136] C. Martinez-Ortiz, M. Koolen, F. Buschenhenke, V. Dalen-Oskam, et al. Beyond the book: Linking books to wikipedia. In *e-Science (e-Science), 2015 IEEE 11th International Conference on*, pages 12–21. IEEE, 2015. (Cited on page 143.)

[137] P. Massa and F. Scrinzi. Manypedia: Comparing language points of view of wikipedia communities. In *Proceedings of the Eighth Annual International Symposium on Wikis and Open Collaboration*, WikiSym '12, pages 21:1–21:9, New York, NY, USA, 2012. ACM. (Cited on page 23.)

[138] Q. Mei, D. Cai, D. Zhang, and C. Zhai. Topic modeling with network regularization. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 101–110, New York, NY, USA, 2008. ACM. (Cited on page 30.)

[139] E. Meij, M. Bron, L. Hollink, B. Huurnink, and M. de Rijke. Learning semantic query suggestions. In A. Bernstein, D. R. Karger, T. Heath, L. Feigenbaum, D. Maynard, E. Motta, and K. Thirunarayan, editors, *8th International Semantic Web Conference*, ISWC 2009, pages 424–440, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. (Cited on pages 30, 89, and 90.)

[140] E. Meij, M. Bron, L. Hollink, B. Huurnink, and M. de Rijke. Mapping queries to the linking open data cloud: A case study using DBpedia. *Journal of Web Semantics*, 9(4):418–433, 2011. (Cited on pages 29 and 89.)

[141] E. Meij, W. Weerkamp, and M. de Rijke. Adding semantics to microblog posts. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 563–572, New York, NY, USA, 2012. ACM. (Cited on pages 29, 30, 90, 94, 95, 96, 100, and 104.)

[142] E. Meij, K. Balog, and D. Odijk. Entity linking and retrieval. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 1127–1127, New York, NY, USA, 2013. ACM. (Cited on page 12.)

[143] E. Meij, K. Balog, and D. Odijk. Entity linking and retrieval for semantic search. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 683–684, New York, NY, USA, 2014. ACM. (Cited on page 12.)

[144] M. Meijer and J. Kleinnijenhuis. Issue news and corporate reputation: Applying the theories of agenda setting and issue ownership in the field of business communication. *Journal of Communication*, 56(3): 543–559, 2006. (Cited on pages 23 and 53.)

[145] D. Metzler and W. Bruce Croft. Linear feature-based models for information retrieval. *Information Retrieval*, 10(3):257–274, 2007. (Cited on pages 101 and 106.)

[146] D. Metzler and W. B. Croft. Combining the language model and inference network approaches to retrieval. *Information Processing & Management*, 40(5):735–750, 2004. (Cited on page 124.)

[147] J. Michel, Y. Shen, A. Aiden, A. Veres, M. Gray, J. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, et al. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176, 2011. (Cited on page 20.)

[148] R. Mihalcea and A. Csomai. Wikify!: Linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, pages 233–242, New York, NY, USA, 2007. ACM. (Cited on pages 29 and 89.)

[149] D. Milne and I. H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 509–518, New York, NY, USA, 2008. ACM. (Cited on pages 29, 30, 89, 95, 102, 149, and 151.)

[150] A. Moffat and J. Zobel. Rank-biased precision for measurement of retrieval effectiveness. *ACM Transactions on Information Systems*, 27(1):2:1–2:27, Dec. 2008. (Cited on page 19.)

[151] A. Mohan, Z. Chen, and K. Q. Weinberger. Web-search ranking with initialized gradient boosted regression trees. *JMLR: Workshop and Conference Proceedings*, 14:77–89, 2011. (Cited on page 100.)

[152] C. Monz, V. Nastase, M. Negri, A. Fahrni, Y. Mehdad, and M. Strube. Cosyne: A framework for multilingual content synchronization of wikis. In *Proceedings of the 7th International Symposium on Wikis and Open Collaboration*, WikiSym '11, pages 217–218, New York, NY, USA, 2011. ACM. (Cited on page 23.)

[153] N. Moraveji, D. Russell, J. Bien, and D. Mease. Measuring improvement in user search performance resulting from optimal search tips. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 355–364, New York, NY, USA, 2011. ACM. (Cited on pages 27 and 142.)

[154] W. R. Neuman, M. R. Just, and A. N. Crigler. *Common knowledge: News and the construction of political meaning*. University of Chicago Press, 1992. (Cited on pages 24 and 54.)

[155] Nielsen. In the U.S., tablets are TV buddies while ereaders make great bedfellows, May 2012. `http://bit.ly/L41f9E` [Online; accessed May 2012]. (Cited on pages 5, 28, and 89.)

[156] Nielsen. State of the media: The cross-platform report. `http://bit.ly/1hNXs4m` [Online; accessed Feb 2014], 2013. (Cited on pages 5, 27, and 113.)

[157] M. C. Nisbet and M. Huge. Attention cycles and frames in the plant biotechnology debate managing power and participation through the press/policy connection. *The Harvard International Journal of Press/Politics*, 11(2):3–40, 2006. (Cited on pages 25 and 55.)

[158] N. F. Noy and M. A. Musen. Prompt: Algorithm and tool for automated ontology merging and alignment. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 450–455. AAAI Press, 2000. (Cited on page 23.)

[159] D. Odijk, O. de Rooij, M.-H. Peetz, T. Pieters, M. de Rijke, and S. Snelders. Semantic document selection: Historical research on collections that span multiple centuries. In *Proceedings of the Second International Conference on Theory and Practice of Digital Libraries*, TPDL 2012, pages 215–221, Berlin, Heidelberg, 2012. Springer-Verlag. (Cited on pages 11, 12, 22, and 37.)

[160] D. Odijk, G. Santucci, M. de Rijke, M. Angelini, and G. L. Granato. Time-aware exploratory search: Exploring word meaning through time. In *SIGIR 2012 Workshop on Time-aware Information Access (#TAIA2012)*, 2012. (Cited on pages 12 and 51.)

[161] D. Odijk, B. Burscher, R. Vliegenthart, and M. de Rijke. Automatic thematic content analysis: Finding frames in news. In *Proceedings of the 5th International Conference on Social Informatics - Volume 8238*, SocInfo 2013, pages 333–345, New York, NY, USA, 2013. Springer-Verlag New York, Inc. (Cited on page 11.)

[162] D. Odijk, E. Meij, and M. de Rijke. Feeding the second screen: Semantic linking based on subtitles. In *Proceedings of the 10th Conference on Open Research Areas in Information Retrieval*, OAIR '13, pages 9–16, Paris, France, France, 2013. Le Centre de Hautes Etudes Internationales d'Informatique Documentaire. (Cited on pages 11, 12, and 30.)

[163] D. Odijk, E. Meij, D. Graus, and T. Kenter. Multilingual semantic linking for video streams: Making "ideas worth sharing" more accessible. In *WWW2013 Workshop on Web of Linked Entities*, 2013. (Cited on page 12.)

[164] D. Odijk, C. Gârbacea, T. Schoegje, L. Hollink, V. Boer, K. Ribbens, and J. Ossenbruggen. Supporting exploration of historical perspectives across collections. In S. Kapidakis, C. Mazurek, and M. Werla,

editors, *Research and Advanced Technology for Digital Libraries: 19th International Conference on Theory and Practice of Digital Libraries*, TPDL 2015, pages 238–251. Springer International Publishing, 2015. (Cited on pages 11 and 12.)

[165] D. Odijk, E. Meij, I. Sijaranamual, and M. de Rijke. Dynamic query modeling for related content finding. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 33–42, New York, NY, USA, 2015. ACM. (Cited on pages 11 and 12.)

[166] D. Odijk, R. W. White, A. Hassan Awadallah, and S. T. Dumais. Struggling and success in web search. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 1551–1560, New York, NY, USA, 2015. ACM. (Cited on pages 1 and 11.)

[167] D. Odijk, E. Meij, and M. de Rijke. Real-time entity linking based on subtitles. *Under Review*, 2016. (Cited on pages 11 and 12.)

[168] R. Oliver. *Satisfaction: A Behavioral Perspective on the Consumer*. ME Sharpe, 2011. (Cited on page 25.)

[169] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Techn. report, Stanford InfoLab, 1999. (Cited on pages 16, 30, and 99.)

[170] R. M. Palau and M.-F. Moens. Argumentation mining: The detection, classification and structure of arguments in text. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, ICAIL '09, pages 98–107, New York, NY, USA, 2009. ACM. (Cited on pages 20 and 53.)

[171] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008. (Cited on pages 20 and 53.)

[172] E. A. Parsons. *The Alexandrian Library: Glory of the Hellenic World, Its Rise, Antiquities, and Destruction*. Elsevier Press, 1952. (Cited on page 17.)

[173] M.-H. Peetz and M. de Rijke. Cognitive temporal document priors. In P. Serdyukov, P. Braslavski, S. O. Kuznetsov, J. Kamps, S. Rüger, E. Agichtein, I. Segalovich, and E. Yilmaz, editors, *Advances in Information Retrieval: 35th European Conference on IR Research*, ECIR 2013, pages 318–330, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. (Cited on pages 43 and 132.)

[174] M.-H. Peetz, E. Meij, M. de Rijke, and W. Weerkamp. Adaptive temporal query modeling. In R. Baeza-Yates, A. P. Vries, H. Zaragoza, B. B. Cambazoglu, V. Murdock, R. Lempel, and F. Silvestri, editors, *Advances in Information Retrieval: 34th European Conference on IR Research*, ECIR 2012, pages 455–458, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. (Cited on page 17.)

[175] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 275–281, New York, NY, USA, 1998. ACM. (Cited on page 15.)

[176] J. Pustejovsky, J. M. Castano, R. Ingria, R. Sauri, R. J. Gaizauskas, A. Setzer, G. Katz, and D. R. Radev. Timeml: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34, 2003. (Cited on page 42.)

[177] F. Radlinski, M. Kurup, and T. Joachims. How does clickthrough data reflect retrieval quality? In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 43–52, New York, NY, USA, 2008. ACM. (Cited on pages 117 and 122.)

[178] K. Raman, P. N. Bennett, and K. Collins-Thompson. Toward whole-session relevance: Exploring intrinsic diversity in web search. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 463–472, New York, NY, USA, 2013. ACM. (Cited on pages 88 and 141.)

[179] L. Ratinov, D. Downey, M. Anderson, and D. Roth. Local and global algorithms for disambiguation to Wikipedia. In *ACL '11*, pages 1375–1384. ACL, 2011. (Cited on pages 29 and 30.)

[180] Razorfish. Forget mobile, think multiscreen. `http://razorfish-outlook.razorfish.com/articles/forgetmobile.aspx`, 2011. (Cited on pages 5, 27, 28, and 113.)

[181] R. Reinanda, D. Odijk, and M. de Rijke. Exploring entity associations over time. In *SIGIR 2013 Workshop on Time-aware Information Access*, 2013. (Cited on page 12.)

[182] Z. Ren, S. Liang, E. Meij, and M. de Rijke. Personalized time-aware tweets summarization. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 513–522, New York, NY, USA, 2013. ACM. (Cited on pages 17, 133, and 142.)

[183] C. Roberts. *Text analysis for the social sciences: Methods for drawing statistical inferences from texts and transcripts*. Lawrence Erlbaum, New York, 1997. (Cited on pages 24 and 54.)

[184] S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. *NIST special publication*, (500225):109–123, 1995. (Cited on page 15.)

[185] S. E. Robertson. The probability ranking principle in ir. *Journal of Documentation*, 33(4):294–304, 1977. (Cited on page 15.)

[186] N. Ruigrok and W. Van Atteveldt. Global angling with a local angle: How US, British, and Dutch newspapers frame global and local terrorist attacks. *The Harvard International Journal of Press/Politics*, 12(1):68–90, 2007. (Cited on pages 24 and 54.)

[187] G. Salton. *Automatic Information Organization and Retrieval.* McGraw Hill Text, 1968. (Cited on page 13.)

[188] G. Salton. Mathematics and information retrieval. *Journal of Documentation*, 35(1):1–29, 1979. (Cited on page 15.)

[189] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, Nov. 1975. (Cited on page 15.)

[190] G. Salton, E. A. Fox, and H. Wu. Extended boolean information retrieval. *Communications of the ACM*, 26(11):1022–1036, Nov. 1983. (Cited on page 15.)

[191] D. Savenkov and E. Agichtein. To hint or not: Exploring the effectiveness of search hints for complex informational tasks. In *Proceedings of the 37th International ACM SIGIR Conference on Research &#38; Development in Information Retrieval*, SIGIR '14, pages 1115–1118, New York, NY, USA, 2014. ACM. (Cited on page 27.)

[192] A. T. Scaria, R. M. Philip, R. West, and J. Leskovec. The last click: Why users give up information network navigation. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 213–222, New York, NY, USA, 2014. ACM. (Cited on pages 26 and 88.)

[193] D. A. Scheufele. Framing as a theory of media effects. *Journal of Communication*, 49(1):103–122, 1999. (Cited on pages 24 and 54.)

[194] G. Schreiber, A. Amin, M. Assem, V. Boer, L. Hardman, M. Hildebrand, L. Hollink, Z. Huang, J. Kersen, M. Niet, B. Omelayenko, J. Ossenbruggen, R. Siebes, J. Taekema, J. Wielemaker, and B. Wielinga. Multimedian e-culture demonstrator. In I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. M. Aroyo, editors, *5th International Semantic Web Conference*, ISWC 2006, pages 951–958, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. (Cited on page 23.)

[195] S. Schreibman, R. Siemens, and J. Unsworth. *A companion to digital humanities*. John Wiley & Sons, 2008. (Cited on pages 20 and 35.)

[196] D. Sculley. Combined regression and ranking. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 979–988, New York, NY, USA, 2010. ACM. (Cited on page 56.)

[197] H. A. Semetko and P. M. Valkenburg. Framing european politics: A content analysis of press and television news. *Journal of Communication*, 50(2):93–109, 2000. (Cited on pages 24, 54, 57, and 58.)

[198] D. V. Shah, M. D. Watts, D. Domke, and D. P. Fan. News framing and cueing of issue regimes: Explaining clinton's public approval in spite of scandal. *Public Opinion Quarterly*, 66(3):339–370, 2002. (Cited on pages 24 and 54.)

[199] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 807–814, New York, NY, USA, 2007. ACM. (Cited on page 56.)

[200] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *IEEE Symposium on Visual Languages*, pages 336–343. IEEE Computer Society, 1996. (Cited on page 22.)

[201] M. Shokouhi, R. W. White, P. Bennett, and F. Radlinski. Fighting search engine amnesia: Reranking repeated results. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 273–282, New York, NY, USA, 2013. ACM. (Cited on pages 1, 25, 52, and 67.)

[202] A. Simon and M. Xenos. Media framing and effective public deliberation. *Political Communication*, 17 (4):363–376, 2000. (Cited on pages 4, 24, 53, 54, and 137.)

[203] C. G. M. Snoek, K. E. A. van de Sande, O. de Rooij, B. Huurnink, E. Gavves, D. Odijk, M. de Rijke, T. Gevers, M. Worring, D. C. Koelma, et al. The MediaMill TRECVID 2010 semantic video search engine. In *Proceedings of the 8th TRECVID Workshop*, 2010. (Cited on page 12.)

[204] H.-J. Song, J. Go, S.-B. Park, and S.-Y. Park. A just-in-time keyword extraction from meeting transcripts. In *NAACL-HLT '13*, pages 888–896, 2013. (Cited on page 28.)

[205] W. Stock and M. Stock. *Handbook of Information Science*. Berlin, Boston, MA: De Gruyter Saur, 2013. (Cited on page 14.)

[206] J. Strötgen and M. Gertz. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2):269–298, 2012. (Cited on pages 41 and 42.)

[207] R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*. MIT Press, 1998. (Cited on

pages 16, 115, and 117.)

[208] G. Tassey, B. R. Rowe, D. W. Wood, A. N. Link, and D. A. Simoni. Economic impact assessment of NIST's Text REtrieval Conference (TREC) program. Techn. report, RTI International, 2010. (Cited on page 17.)

[209] J. Teevan. The re: search engine: simultaneous support for finding and re-finding. In *UIST'07*, pages 23–32, 2007. (Cited on pages 27 and 69.)

[210] K. Toutanova, C. D. Manning, and A. Y. Ng. Learning random walk models for inducing word dependency distributions. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 103–, New York, NY, USA, 2004. ACM. (Cited on page 30.)

[211] M. C. Traub and J. van Ossenbruggen. Workshop on tool criticism in the digital humanities. Techn. report, CWI, 2015. (Cited on page 141.)

[212] M. Tsagkias, M. de Rijke, and W. Weerkamp. Hypergeometric language models for republished article finding. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 485–494, New York, NY, USA, 2011. ACM. (Cited on pages 17, 133, and 142.)

[213] J. Uijlings, O. de Rooij, D. Odijk, A. Smeulders, and M. Worring. Instant bag-of-words served on a laptop. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, ICMR '11, pages 69:1–69:2, New York, NY, USA, 2011. ACM. (Cited on page 12.)

[214] C. van Rijsbergen. *Information Retrieval*. Butterwoths, 1979. (Cited on page 18.)

[215] F. van Vree. *De Nederlandse pers en Duitsland 1930–1939*. Historische Uitgeverij, 1989. (Cited on pages 21, 35, and 36.)

[216] H. Varian. Why data matters. `https://googleblog.blogspot.nl/2008/03/why-data-matters.html` [Online, accessed December 2015], March 2008. (Cited on page 17.)

[217] P. Vaswani and J. Cameron. The national physical laboratory experiments in statistical word associations and their use in document indexing and retrieval. Techn. report, ERIC, 1970. (Cited on page 17.)

[218] R. Vliegenthart, H. G. Boomgaarden, and J. W. Boumans. *Changes in political news coverage*. Palgrave Macmillan, 2011. (Cited on pages 24 and 54.)

[219] E. M. Voorhees and D. Harman, editors. *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*, 2000. (Cited on page 17.)

[220] E. M. Voorhees, D. K. Harman, et al. *TREC: Experiment and evaluation in information retrieval*, volume 1. MIT press Cambridge, 2005. (Cited on page 17.)

[221] N. Voskarides, D. Odijk, M. Tsagkias, W. Weerkamp, and M. de Rijke. Query-dependent contextualization of streaming data. In M. de Rijke, T. Kenter, A. P. de Vries, C. Zhai, F. de Jong, K. Radinsky, and K. Hofmann, editors, *Advances in Information Retrieval: 36th European Conference on IR Research*, ECIR 2014, pages 706–712. Springer International Publishing, 2014. (Cited on pages 12 and 143.)

[222] H. Wang, Y. Song, M.-W. Chang, X. He, A. Hassan, and R. W. White. Modeling action-level satisfaction for search task satisfaction prediction. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, pages 123–132, New York, NY, USA, 2014. ACM. (Cited on page 26.)

[223] C. J. C. H. Watkins. *Learning from delayed rewards*. PhD thesis, University of Cambridge, 1989. (Cited on page 115.)

[224] W. Weaver. Translation. *Machine Translation of Languages*, 14:15–23, 1955. (Cited on page 20.)

[225] R. W. White and S. T. Dumais. Characterizing and predicting search engine switching behavior. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 87–96, New York, NY, USA, 2009. ACM. (Cited on page 27.)

[226] R. W. White and D. Morris. Investigating the querying and browsing behavior of advanced search engine users. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 255–262, New York, NY, USA, 2007. ACM. (Cited on pages 15, 27, 70, 88, and 142.)

[227] R. W. White, M. Bilenko, and S. Cucerzan. Studying the use of popular destinations to enhance web search interaction. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 159–166, New York, NY, USA, 2007. ACM. (Cited on page 26.)

[228] R. W. White, S. T. Dumais, and J. Teevan. Characterizing the influence of domain expertise on web search behavior. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 132–141, New York, NY, USA, 2009. ACM. (Cited on pages 2, 25, 26, 52, 67, 69, 88, and 142.)

[229] E. B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American*

Statistical Association, 22(158):209–212, 1927. (Cited on pages 95 and 149.)

[230] E. Witte. *De constructie van België: 1828–1847*. Lannoo Uitgeverij, 2006. (Cited on pages 21, 35, and 36.)

[231] Y. Xu and D. Mease. Evaluating web search using task completion time. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 676–677, New York, NY, USA, 2009. ACM. (Cited on page 26.)

[232] E. Yom-Tov, S. Dumais, and Q. Guo. Promoting civil discourse through search engine diversity. *Social Science Computer Review*, 32(2):145–154, 2013. (Cited on page 64.)

[233] Y. Yue and T. Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1201–1208, New York, NY, USA, 2009. ACM. (Cited on pages 16, 117, 121, and 122.)

[234] J. Zhang. *Visualization for Information Retrieval*. Springer, Berlin, Heidelberg, 2008. (Cited on pages 3 and 22.)

[235] D. Zillmann and H. B. Brosius. *Exemplification in communication*. Hogrefe and Huber, 2000. (Cited on page 24.)

Samenvatting

Dit proefschrift beschrijft onderzoek naar een kerndoel van *information retrieval (IR)*: gebruikers gemakkelijk toegang geven tot informatie. Aan de hand van drie onderzoeksthemas behandelt het onderzoek drie aspecten van IR: het domein waarin een IR-systeem wordt gebruikt, de interactie van gebruikers met het systeem, en het scenario voor hoe deze gebruikers met het systeem werken. Centraal in deze onderzoeksthema's is het streven om inzicht te krijgen in het gedrag van gebruikers van zoekmachines en algoritmes te ontwikkelen om hen te ondersteunen in hun zoektocht, of het nu gaat om een een onderzoeker die een grote collectie verkent of bestudeert, iemand die zoekt op het web en moeite heeft om iets te vinden, of een televisiekijker op zoek naar achtergrondinformatie.

Het eerste onderzoeksthema is gemotiveerd door de zoektaken van onderzoekers bij het verkennen en bestuderen van grote collecties, zoals een krantenarchief. Om hun zoektocht op grotere schaal mogelijk te maken stellen wij computationele technieken voor om collecties met elkaar te verbinden en om te bepalen vanuit welk perspectief een nieuwsbericht is geschreven. Gemotiveerd door de manier waarop historici documenten selecteren voor nauwkeurige lezing stellen wij nieuwe technieken voor om verbindingen te leggen tussen collecties op basis van automatisch geëxtraheerde verwijzingen naar een tijdsperiode. Om te illustreren hoe deze algoritmes gebruikt kunnen worden introduceren we een zoekinterface voor het verkennen en analyseren van de nieuw verbonden collecties. Deze interface belicht verschillende perspectieven en vereist weinig domeinkennis. Gebaseerd op hoe communicatiewetenschappers *framing* in nieuws bestuderen stellen wij een automatische thematische content analyse benadering voor.

Het tweede onderzoeksthema wordt behandeld in een multi-methodologisch onderzoek naar hoe gebruikers van een web zoekmachine zich gedragen als ze niet kunnen vinden wat ze zoeken. Op basis van grootschalige log analyse, annotaties op basis van *crowdsourcing*, en voorspellend modelleren laten we zien dat het gedrag van gebruikers samenhangt met hoe succesvol zij uiteindelijk zijn. Gebaseerd op deze bevindingen doen we voorstellen hoe een systeem frustratie van gebruikers in hun zoektocht kan voorkomen. Om gebruikers te ondersteunen stellen we algoritmes voor die de aard van toekomstige acties en de verwachte impact op de zoekuitkomst accuraat voorspellen. We doen aanbevelingen voor het ontwerp van zoeksystemen om de frustratie van gebruikers te beperken en om hen uiteindelijk succesvoller te laten zoeken.

In het derde en laatste onderzoeksthema kijken we naar een proactief zoekscenario in een live televisie setting. We stellen algoritmes voor die contextuele informatie kunnen gebruiken om automatisch divers gerelateerd materiaal te vinden voor een achteroverleunende televisiekijker. Terwijl ze tv kijken, consumeren mensen steeds meer aanvullend materiaal, gerelateerd aan wat ze kijken. We introduceren twee technieken om automatisch materiaal te vinden op basis van ondertiteling: één gebaseerd op *entity linking* en één die op basis van *reinforcement learning* effectieve zoekopdrachten naar gerelateerd materiaal genereert. Beide technieken zijn uitermate efficiënt en worden op dit moment gebruikt voor live televisie-uitzendingen.

Elk onderzoekshoofdstuk in dit proefschrift levert inzichten en algoritmes op die gebruikers van IR-toepassingen helpen bij het zoeken. Voor verschillende domeinen, gebruikers, en toegangsscenario's verbetert het gepresenteerde onderzoek het gemak van toegang tot informatie.

SIKS Dissertation Series

1998

1 Johan van den Akker (CWI) *DEGAS: An Active, Temporal Database of Autonomous Objects*

2 Floris Wiesman (UM) *Information Retrieval by Graphically Browsing Meta-Information*

3 Ans Steuten (TUD) *A Contribution to the Linguistic Analysis of Business Conversations*

4 Dennis Breuker (UM) *Memory versus Search in Games*

5 E. W. Oskamp (RUL) *Computerondersteuning bij Straftoemeting*

1999

1 Mark Sloof (VUA) *Physiology of Quality Change Modelling: Automated modelling of*

2 Rob Potharst (EUR) *Classification using decision trees and neural nets*

3 Don Beal (UM) *The Nature of Minimax Search*

4 Jacques Penders (UM) *The practical Art of Moving Physical Objects*

5 Aldo de Moor (KUB) *Empowering Communities: A Method for the Legitimate User-Driven*

6 Niek J. E. Wijngaards (VUA) *Re-design of compositional systems*

7 David Spelt (UT) *Verification support for object database design*

8 Jacques H. J. Lenting (UM) *Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism*

2000

1 Frank Niessink (VUA) *Perspectives on Improving Software Maintenance*

2 Koen Holtman (TUe) *Prototyping of CMS Storage Management*

3 Carolien M. T. Metselaar (UvA) *Sociaal-organisatorische gevolgen van kennistechnologie*

4 Geert de Haan (VUA) *ETAG, A Formal Model of Competence Knowledge for User Interface*

5 Ruud van der Pol (UM) *Knowledge-based Query Formulation in Information Retrieval*

6 Rogier van Eijk (UU) *Programming Languages for Agent Communication*

7 Niels Peek (UU) *Decision-theoretic Planning of Clinical Patient Management*

8 Veerle Coupé (EUR) *Sensitivity Analyis of Decision-Theoretic Networks*

9 Florian Waas (CWI) *Principles of Probabilistic Query Optimization*

10 Niels Nes (CWI) *Image Database Management System Design Considerations, Algorithms and Architecture*

11 Jonas Karlsson (CWI) *Scalable Distributed Data Structures for Database Management*

2001

1 Silja Renooij (UU) *Qualitative Approaches to Quantifying Probabilistic Networks*

2 Koen Hindriks (UU) *Agent Programming Languages: Programming with Mental Models*

3 Maarten van Someren (UvA) *Learning as problem solving*

4 Evgueni Smirnov (UM) *Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets*

5 Jacco van Ossenbruggen (VUA) *Processing Structured Hypermedia: A Matter of Style*

6 Martijn van Welie (VUA) *Task-based User Interface Design*

7 Bastiaan Schonhage (VUA) *Diva: Architectural Perspectives on Information Visualization*

8 Pascal van Eck (VUA) *A Compositional Semantic Structure for Multi-Agent Systems Dynamics*

9 Pieter Jan 't Hoen (RUL) *Towards Distributed Development of Large Object-Oriented Models*

10 Maarten Sierhuis (UvA) *Modeling and Simulating Work Practice*

11 Tom M. van Engers (VUA) *Knowledge Management*

2002

1 Nico Lassing (VUA) *Architecture-Level Modifiability Analysis*

2 Roelof van Zwol (UT) *Modelling and searching web-based document collections*

3 Henk Ernst Blok (UT) *Database Optimization Aspects for Information Retrieval*

4 Juan Roberto Castelo Valdueza (UU) *The Discrete Acyclic Digraph Markov Model in Data Mining*

5 Radu Serban (VUA) *The Private Cyberspace Modeling Electronic*

6 Laurens Mommers (UL) *Applied legal epistemology: Building a knowledge-based ontology of*

7 Peter Boncz (CWI) *Monet: A Next-Generation DBMS Kernel For Query-Intensive*

8 Jaap Gordijn (VUA) *Value Based Requirements Engineering: Exploring Innovative*

9 Willem-Jan van den Heuvel (KUB) *Integrating Modern Business Applications with Objectified Legacy*

10 Brian Sheppard (UM) *Towards Perfect Play of Scrabble*

11 Wouter C. A. Wijngaards (VUA) *Agent Based Modelling of Dynamics: Biological and Organisational Applications*

12 Albrecht Schmidt (UvA) *Processing XML in Database Systems*

13 Hongjing Wu (TUe) *A Reference Architecture for Adaptive Hypermedia Applications*

14 Wieke de Vries (UU) *Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems*

15 Rik Eshuis (UT) *Semantics and Verification of UML Activity Diagrams for Workflow Modelling*

16 Pieter van Langen (VUA) *The Anatomy of Design: Foundations, Models and Applications*

17 Stefan Manegold (UvA) *Understanding, Modeling, and Improving Main-Memory Database Performance*

2003

1 Heiner Stuckenschmidt (VUA) *Ontology-Based Information Sharing in Weakly Structured Environments*

2 Jan Broersen (VUA) *Modal Action Logics for Reasoning About Reactive Systems*

3 Martijn Schuemie (TUD) *Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy*

4 Milan Petkovic (UT) *Content-Based Video Retrieval Supported by Database Technology*

5 Jos Lehmann (UvA) *Causation in Artificial Intelligence and Law: A modelling approach*

6 Boris van Schooten (UT) *Development and specification of virtual environments*

7 Machiel Jansen (UvA) *Formal Explorations of Knowledge Intensive Tasks*

8 Yongping Ran (UM) *Repair Based Scheduling*

9 Rens Kortmann (UM) *The resolution of visually guided behaviour*

10 Andreas Lincke (UvT) *Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture*

11 Simon Keizer (UT) *Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks*

12 Roeland Ordelman (UT) *Dutch speech recognition in multimedia information retrieval*

13 Jeroen Donkers (UM) *Nosce Hostem: Searching with Opponent Models*

14 Stijn Hoppenbrouwers (KUN) *Freezing Language: Conceptualisation Processes across ICT-Supported Organisations*

15 Mathijs de Weerdt (TUD) *Plan Merging in Multi-Agent Systems*

16 Menzo Windhouwer (CWI) *Feature Grammar Systems: Incremental Maintenance of Indexes to Digital Media Warehouses*

17 David Jansen (UT) *Extensions of Statecharts with Probability, Time, and Stochastic Timing*

18 Levente Kocsis (UM) *Learning Search Decisions*

2004

1 Virginia Dignum (UU) *A Model for Organizational Interaction: Based on Agents, Founded in Logic*

2 Lai Xu (UvT) *Monitoring Multi-party Contracts for E-business*

3 Perry Groot (VUA) *A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving*

4 Chris van Aart (UvA) *Organizational Principles for Multi-Agent Architectures*

5 Viara Popova (EUR) *Knowledge discovery and monotonicity*

6 Bart-Jan Hommes (TUD) *The Evaluation of Business Process Modeling Techniques*

7 Elise Boltjes (UM) *Voorbeeldig onderwijs: voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes*

8 Joop Verbeek (UM) *Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politiële gegevensuitwisseling en digitale expertise*

9 Martin Caminada (VUA) *For the Sake of the Argument: explorations into argument-based reasoning*

10 Suzanne Kabel (UvA) *Knowledge-rich indexing of learning-objects*

11 Michel Klein (VUA) *Change Management for Distributed Ontologies*

12 The Duy Bui (UT) *Creating emotions and facial expressions for embodied agents*

13 Wojciech Jamroga (UT) *Using Multiple Models of Reality: On Agents who Know how to Play*

14 Paul Harrenstein (UU) *Logic in Conflict. Logical Explorations in Strategic Equilibrium*

15 Arno Knobbe (UU) *Multi-Relational Data Mining*

16 Federico Divina (VUA) *Hybrid Genetic Relational Search for Inductive Learning*

17 Mark Winands (UM) *Informed Search in Complex Games*

18 Vania Bessa Machado (UvA) *Supporting the Construction of Qualitative Knowledge Models*

19 Thijs Westerveld (UT) *Using generative probabilistic models for multimedia retrieval*

20 Madelon Evers (Nyenrode) *Learning from Design: facilitating multidisciplinary design teams*

2005

1 Floor Verdenius (UvA) *Methodological Aspects of Designing Induction-Based Applications*

2 Erik van der Werf (UM) *AI techniques for the game of Go*

3 Franc Grootjen (RUN) *A Pragmatic Approach to the Conceptualisation of Language*

4 Nirvana Meratnia (UT) *Towards Database Support for Moving Object data*

5 Gabriel Infante-Lopez (UvA) *Two-Level Probabilistic Grammars for Natural Language Parsing*

6 Pieter Spronck (UM) *Adaptive Game AI*

7 Flavius Frasincar (TUe) *Hypermedia Presentation Generation for Semantic Web Information Systems*

8 Richard Vdovjak (TUe) *A Model-driven Approach for Building Distributed Ontology-based Web Applications*

9 Jeen Broekstra (VUA) *Storage, Querying and Inferencing for Semantic Web Languages*

10 Anders Bouwer (UvA) *Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments*

11 Elth Ogston (VUA) *Agent Based Matchmaking and Clustering: A Decentralized Approach to Search*

12 Csaba Boer (EUR) *Distributed Simulation in Industry*

13 Fred Hamburg (UL) *Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen*

14 Borys Omelayenko (VUA) *Web-Service configuration on the Semantic Web: Exploring how semantics meets pragmatics*

15 Tibor Bosse (VUA) *Analysis of the Dynamics of Cognitive Processes*

16 Joris Graaumans (UU) *Usability of XML Query Languages*

17 Boris Shishkov (TUD) *Software Specification Based on Re-usable Business Components*

18 Danielle Sent (UU) *Test-selection strategies for probabilistic networks*

19 Michel van Dartel (UM) *Situated Representation*

20 Cristina Coteanu (UL) *Cyber Consumer Law, State of the Art and Perspectives*

21 Wijnand Derks (UT) *Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics*

2006

1 Samuil Angelov (TUe) *Foundations of B2B Electronic Contracting*

2 Cristina Chisalita (VUA) *Contextual issues in the design and use of information technology in organizations*

3 Noor Christoph (UvA) *The role of metacognitive skills in learning to solve problems*

4 Marta Sabou (VUA) *Building Web Service Ontologies*

5 Cees Pierik (UU) *Validation Techniques for Object-Oriented Proof Outlines*

6 Ziv Baida (VUA) *Software-aided Service Bundling: Intelligent Methods & Tools for Graphical Service Modeling*

7 Marko Smiljanic (UT) *XML schema matching: balancing efficiency and effectiveness by means of clustering*

8 Eelco Herder (UT) *Forward, Back and Home Again: Analyzing User Behavior on the Web*

9 Mohamed Wahdan (UM) *Automatic Formulation of the Auditor's Opinion*

10 Ronny Siebes (VUA) *Semantic Routing in Peer-to-Peer Systems*

11 Joeri van Ruth (UT) *Flattening Queries over Nested Data Types*

12 Bert Bongers (VUA) *Interactivation: Towards an e-cology of people, our technological environment, and the arts*

13 Henk-Jan Lebbink (UU) *Dialogue and Decision Games for Information Exchanging Agents*

14 Johan Hoorn (VUA) *Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change*

15 Rainer Malik (UU) *CONAN: Text Mining in the Biomedical Domain*

16 Carsten Riggelsen (UU) *Approximation Methods for Efficient Learning of Bayesian Networks*

17 Stacey Nagata (UU) *User Assistance for Multitasking with Interruptions on a Mobile Device*

18 Valentin Zhizhkun (UvA) *Graph transformation for Natural Language Processing*

19 Birna van Riemsdijk (UU) *Cognitive Agent Programming: A Semantic Approach*

20 Marina Velikova (UvT) *Monotone models for prediction in data mining*

21 Bas van Gils (RUN) *Aptness on the Web*

22 Paul de Vrieze (RUN) *Fundaments of Adaptive Personalisation*

23 Ion Juvina (UU) *Development of Cognitive Model for Navigating on the Web*

24 Laura Hollink (VUA) *Semantic Annotation for Retrieval of Visual Resources*

25 Madalina Drugan (UU) *Conditional log-likelihood MDL and Evolutionary MCMC*

26 Vojkan Mihajlovic (UT) *Score Region Algebra: A Flexible Framework for Structured Information Retrieval*

27 Stefano Bocconi (CWI) *Vox Populi: generating video documentaries from semantically annotated media repositories*

28 Borkur Sigurbjornsson (UvA) *Focused Information Access using XML Element Retrieval*

2007

1 Kees Leune (UvT) *Access Control and Service-Oriented Architectures*

2 Wouter Teepe (RUG) *Reconciling Information Exchange and Confidentiality: A Formal Approach*

3 Peter Mika (VUA) *Social Networks and the Semantic Web*

4 Jurriaan van Diggelen (UU) *Achieving Semantic Interoperability in Multi-agent Systems: a dialogue-based approach*

5 Bart Schermer (UL) *Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance*

6 Gilad Mishne (UvA) *Applied Text Analytics for Blogs*

7 Natasa Jovanovic' (UT) *To Whom It May Concern: Addressee Identification in Face-to-Face Meetings*

8 Mark Hoogendoorn (VUA) *Modeling of Change in Multi-Agent Organizations*

9 David Mobach (VUA) *Agent-Based Mediated Service Negotiation*

10 Huib Aldewereld (UU) *Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols*

11 Natalia Stash (TUe) *Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System*

12 Marcel van Gerven (RUN) *Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty*

13 Rutger Rienks (UT) *Meetings in Smart Environments: Implications of Progressing Technology*

14 Niek Bergboer (UM) *Context-Based Image Analysis*

15 Joyca Lacroix (UM) *NIM: a Situated Computational Memory Model*

16 Davide Grossi (UU) *Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems*

17 Theodore Charitos (UU) *Reasoning with Dynamic Networks in Practice*

18 Bart Orriens (UvT) *On the development an management of adaptive business collaborations*

19 David Levy (UM) *Intimate relationships with artificial partners*

20 Slinger Jansen (UU) *Customer Configuration Updating in a Software Supply Network*

21 Karianne Vermaas (UU) *Fast diffusion and broadening use: A research on residential adoption and usage of broadband internet in the Netherlands between 2001 and 2005*

22 Zlatko Zlatev (UT) *Goal-oriented design of value and process models from patterns*

23 Peter Barna (TUe) *Specification of Application Logic in Web Information Systems*

24 Georgina Ramírez Camps (CWI) *Structural Features in XML Retrieval*

25 Joost Schalken (VUA) *Empirical Investigations in Software Process Improvement*

2008

1 Katalin Boer-Sorbán (EUR) *Agent-Based Simulation of Financial Markets: A modular, continuous-time approach*

2 Alexei Sharpanskykh (VUA) *On Computer-Aided Methods for Modeling and Analysis of Organizations*

3 Vera Hollink (UvA) *Optimizing hierarchical menus: a usage-based approach*

4 Ander de Keijzer (UT) *Management of Uncertain Data: towards unattended integration*

5 Bela Mutschler (UT) *Modeling and simulating causal dependencies on process-aware information systems from a cost perspective*

6 Arjen Hommersom (RUN) *On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective*

7 Peter van Rosmalen (OU) *Supporting the tutor in the design and support of adaptive e-learning*

8 Janneke Bolt (UU) *Bayesian Networks: Aspects of Approximate Inference*

9 Christof van Nimwegen (UU) *The paradox of the guided user: assistance can be counter-effective*

10 Wauter Bosma (UT) *Discourse oriented summarization*

11 Vera Kartseva (VUA) *Designing Controls for Network Organizations: A Value-Based Approach*

12 Jozsef Farkas (RUN) *A Semiotically Oriented Cognitive Model of Knowledge Representation*

13 Caterina Carraciolo (UvA) *Topic Driven Access to Scientific Handbooks*

14 Arthur van Bunningen (UT) *Context-Aware Querying: Better Answers with Less Effort*

15 Martijn van Otterlo (UT) *The Logic of Adaptive Behavior: Knowledge Representation and Algorithms for the Markov Decision Process Framework in First-Order Domains*

16 Henriette van Vugt (VUA) *Embodied agents from a user's perspective*

17 Martin Op 't Land (TUD) *Applying Architecture and Ontology to the Splitting and Allying of Enterprises*

18 Guido de Croon (UM) *Adaptive Active Vision*

19 Henning Rode (UT) *From Document to Entity Retrieval: Improving Precision and Performance of Focused Text Search*

20 Rex Arendsen (UvA) *Geen bericht, goed bericht. Een onderzoek naar de effecten van de introductie van elektronisch berichtenverkeer met de overheid op de administratieve lasten van bedrijven*

21 Krisztian Balog (UvA) *People Search in the Enterprise*

22 Henk Koning (UU) *Communication of IT-Architecture*

23 Stefan Visscher (UU) *Bayesian network models for the management of ventilator-associated pneumonia*

24 Zharko Aleksovski (VUA) *Using background knowledge in ontology matching*

25 Geert Jonker (UU) *Efficient and Equitable Exchange in Air Traffic Management Plan Repair using Spender-signed Currency*

26 Marijn Huijbregts (UT) *Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled*

27 Hubert Vogten (OU) *Design and Implementation Strategies for IMS Learning Design*

28 Ildiko Flesch (RUN) *On the Use of Independence Relations in Bayesian Networks*

29 Dennis Reidsma (UT) *Annotations and Subjective Machines: Of Annotators, Embodied Agents, Users, and Other Humans*

30 Wouter van Atteveldt (VUA) *Semantic Network Analysis: Techniques for Extracting, Representing and Querying Media Content*

31 Loes Braun (UM) *Pro-Active Medical Information Retrieval*

32 Trung H. Bui (UT) *Toward Affective Dialogue Management using Partially Observable Markov Decision Processes*

33 Frank Terpstra (UvA) *Scientific Workflow Design: theoretical and practical issues*

34 Jeroen de Knijf (UU) *Studies in Frequent Tree Mining*

35 Ben Torben Nielsen (UvT) *Dendritic morphologies: function shapes structure*

2009

1 Rasa Jurgelenaite (RUN) *Symmetric Causal Independence Models*

2 Willem Robert van Hage (VUA) *Evaluating Ontology-Alignment Techniques*

3 Hans Stol (UvT) *A Framework for Evidence-based Policy Making Using IT*

4 Josephine Nabukenya (RUN) *Improving the Quality of Organisational Policy Making using Collaboration Engineering*

5 Sietse Overbeek (RUN) *Bridging Supply and Demand for Knowledge Intensive Tasks: Based on Knowledge, Cognition, and Quality*

6 Muhammad Subianto (UU) *Understanding Classification*

7 Ronald Poppe (UT) *Discriminative Vision-Based Recovery and Recognition of Human Motion*

8 Volker Nannen (VUA) *Evolutionary Agent-Based Policy Analysis in Dynamic Environments*

9 Benjamin Kanagwa (RUN) *Design, Discovery and Construction of Service-oriented Systems*

10 Jan Wielemaker (UvA) *Logic programming for knowledge-intensive interactive applications*

11 Alexander Boer (UvA) *Legal Theory, Sources of Law & the Semantic Web*

12 Peter Massuthe (TUE, Humboldt-Universitaet zu Berlin) *Operating Guidelines for Services*

13 Steven de Jong (UM) *Fairness in Multi-Agent Systems*

14 Maksym Korotkiy (VUA) *From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA)*

15 Rinke Hoekstra (UvA) *Ontology Representation: Design Patterns and Ontologies that Make Sense*

16 Fritz Reul (UvT) *New Architectures in Computer Chess*

17 Laurens van der Maaten (UvT) *Feature Extraction from Visual Data*

18 Fabian Groffen (CWI) *Armada, An Evolving Database System*

19 Valentin Robu (CWI) *Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets*

20 Bob van der Vecht (UU) *Adjustable Autonomy: Controling Influences on Decision Making*

21 Stijn Vanderlooy (UM) *Ranking and Reliable Classification*

22 Pavel Serdyukov (UT) *Search For Expertise: Going beyond direct evidence*

23 Peter Hofgesang (VUA) *Modelling Web Usage in a Changing Environment*

24 Annerieke Heuvelink (VUA) *Cognitive Models for Training Simulations*

25 Alex van Ballegooij (CWI) *RAM: Array Database Management through Relational Mapping*

26 Fernando Koch (UU) *An Agent-Based Model for the Development of Intelligent Mobile Services*

27 Christian Glahn (OU) *Contextual Support of social Engagement and Reflection on the Web*

28 Sander Evers (UT) *Sensor Data Management with Probabilistic Models*

29 Stanislav Pokraev (UT) *Model-Driven Semantic Integration of Service-Oriented Applications*

30 Marcin Zukowski (CWI) *Balancing vectorized query execution with bandwidth-optimized storage*

31 Sofiya Katrenko (UvA) *A Closer Look at Learning Relations from Text*

32 Rik Farenhorst (VUA) *Architectural Knowledge Management: Supporting Architects and Auditors*

33 Khiet Truong (UT) *How Does Real Affect Affect Affect Recognition In Speech?*

34 Inge van de Weerd (UU) *Advancing in Software Product Management: An Incremental Method Engineering Approach*

35 Wouter Koelewijn (UL) *Privacy en Politiegegevens: Over geautomatiseerde normatieve informatie-uitwisseling*

36 Marco Kalz (OUN) *Placement Support for Learners in Learning Networks*

37 Hendrik Drachsler (OUN) *Navigation Support for Learners in Informal Learning Networks*

38 Riina Vuorikari (OU) *Tags and self-organisation: a metadata ecology for learning resources in a multilingual context*

39 Christian Stahl (TUE, Humboldt-Universitaet zu Berlin) *Service Substitution: A Behavioral Approach Based on Petri Nets*

40 Stephan Raaijmakers (UvT) *Multinomial Language Learning: Investigations into the Geometry of Language*

41 Igor Berezhnyy (UvT) *Digital Analysis of Paintings*

42 Toine Bogers (UvT) *Recommender Systems for Social Bookmarking*

43 Virginia Nunes Leal Franqueira (UT) *Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients*

44 Roberto Santana Tapia (UT) *Assessing Business-IT Alignment in Networked Organizations*

45 Jilles Vreeken (UU) *Making Pattern Mining Useful*

46 Loredana Afanasiev (UvA) *Querying XML: Benchmarks and Recursion*

2010

1 Matthijs van Leeuwen (UU) *Patterns that Matter*

2 Ingo Wassink (UT) *Work flows in Life Science*

3 Joost Geurts (CWI) *A Document Engineering Model and Processing Framework for Multimedia documents*

4 Olga Kulyk (UT) *Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments*

5 Claudia Hauff (UT) *Predicting the Effectiveness of Queries and Retrieval Systems*

6 Sander Bakkes (UvT) *Rapid Adaptation of Video Game AI*

7 Wim Fikkert (UT) *Gesture interaction at a Distance*

8 Krzysztof Siewicz (UL) *Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments*

9 Hugo Kielman (UL) *A Politiele gegevensverwerking en Privacy, Naar een effectieve waarborging*

10 Rebecca Ong (UL) *Mobile Communication and Protection of Children*

11 Adriaan Ter Mors (TUD) *The world according to MARP: Multi-Agent Route Planning*

12 Susan van den Braak (UU) *Sensemaking software for crime analysis*

13 Gianluigi Folino (RUN) *High Performance Data Mining using Bio-inspired techniques*

14 Sander van Splunter (VUA) *Automated Web Service Reconfiguration*

15 Lianne Bodenstaff (UT) *Managing Dependency Relations in Inter-Organizational Models*

16 Sicco Verwer (TUD) *Efficient Identification of Timed Automata, theory and practice*

17 Spyros Kotoulas (VUA) *Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications*

18 Charlotte Gerritsen (VUA) *Caught in the Act: Investigating Crime by Agent-Based Simulation*

19 Henriette Cramer (UvA) *People's Responses to Autonomous and Adaptive Systems*

20 Ivo Swartjes (UT) *Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative*

21 Harold van Heerde (UT) *Privacy-aware data management by means of data degradation*

22 Michiel Hildebrand (CWI) *End-user Support for Access to*
Heterogeneous Linked Data

23 Bas Steunebrink (UU) *The Logical Structure of Emotions*

24 Zulfiqar Ali Memon (VUA) *Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective*

25 Ying Zhang (CWI) *XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines*

26 Marten Voulon (UL) *Automatisch contracteren*

27 Arne Koopman (UU) *Characteristic Relational Patterns*

28 Stratos Idreos (CWI) *Database Cracking: Towards Auto-tuning Database Kernels*

29 Marieke van Erp (UvT) *Accessing Natural History: Discoveries in data cleaning, structuring, and retrieval*

30 Victor de Boer (UvA) *Ontology Enrichment from Heterogeneous Sources on the Web*

31 Marcel Hiel (UvT) *An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems*

32 Robin Aly (UT) *Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval*

33 Teduh Dirgahayu (UT) *Interaction Design in Service Compositions*

34 Dolf Trieschnigg (UT) *Proof of Concept: Concept-based Biomedical Information Retrieval*

35 Jose Janssen (OU) *Paving the Way for Lifelong Learning: Facilitating competence development through a learning path specification*

36 Niels Lohmann (TUe) *Correctness of services and their composition*

37 Dirk Fahland (TUe) *From Scenarios to components*

38 Ghazanfar Farooq Siddiqui (VUA) *Integrative modeling of emotions in virtual agents*

39 Mark van Assem (VUA) *Converting and Integrating Vocabularies for the Semantic Web*

40 Guillaume Chaslot (UM) *Monte-Carlo Tree Search*

41 Sybren de Kinderen (VUA) *Needs-driven service bundling in a multi-supplier setting: the computational e3-service approach*

42 Peter van Kranenburg (UU) *A Computational Approach to Content-Based Retrieval of Folk Song Melodies*

43 Pieter Bellekens (TUe) *An Approach towards Context-sensitive and User-adapted Access to Heterogeneous Data Sources, Illustrated in the Television Domain*

44 Vasilios Andrikopoulos (UvT) *A theory and model for the evolution of software services*

45 Vincent Pijpers (VUA) *e3alignment: Exploring Inter-Organizational Business-ICT Alignment*

46 Chen Li (UT) *Mining Process Model Variants: Challenges, Techniques, Examples*

47 Jahn-Takeshi Saito (UM) *Solving difficult game positions*

48 Bouke Huurnink (UvA) *Search in Audiovisual Broadcast Archives*

49 Alia Khairia Amin (CWI) *Understanding and supporting information seeking tasks in multiple sources*

50 Peter-Paul van Maanen (VUA) *Adaptive Support for Human-Computer Teams: Exploring the Use of Cognitive Models of Trust and Attention*

51 Edgar Meij (UvA) *Combining Concepts and Language Models for Information Access*

2011

1 Botond Cseke (RUN) *Variational Algorithms for Bayesian Inference in Latent Gaussian Models*

2 Nick Tinnemeier (UU) *Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language*

3 Jan Martijn van der Werf (TUe) *Compositional Design and Verification of Component-Based Information Systems*

4 Hado van Hasselt (UU) *Insights in Reinforcement Learning: Formal analysis and empirical evaluation of temporal-difference*

5 Base van der Raadt (VUA) *Enterprise Architecture Coming of Age: Increasing the Performance of an Emerging Discipline*

6 Yiwen Wang (TUe) *Semantically-Enhanced Recommendations in Cultural Heritage*

7 Yujia Cao (UT) *Multimodal Information Presentation for High Load Human Computer Interaction*

8 Nieske Vergunst (UU) *BDI-based Generation of Robust Task-Oriented Dialogues*

9 Tim de Jong (OU) *Contextualised Mobile Media for Learning*

10 Bart Bogaert (UvT) *Cloud Content Contention*

11 Dhaval Vyas (UT) *Designing for Awareness: An Experience-focused HCI Perspective*

12 Carmen Bratosin (TUe) *Grid Architecture for Distributed Process Mining*

13 Xiaoyu Mao (UvT) *Airport under Control. Multiagent Scheduling for Airport Ground Handling*

14 Milan Lovric (EUR) *Behavioral Finance and Agent-Based Artificial Markets*

15 Marijn Koolen (UvA) *The Meaning of Structure: the Value of Link Evidence for Information Retrieval*

16 Maarten Schadd (UM) *Selective Search in Games of Different Complexity*

17 Jiyin He (UvA) *Exploring Topic Structure: Coherence, Diversity and Relatedness*

18 Mark Ponsen (UM) *Strategic Decision-Making in complex games*

19 Ellen Rusman (OU) *The Mind ' s Eye on Personal Profiles*

20 Qing Gu (VUA) *Guiding service-oriented software engineering: A view-based approach*

21 Linda Terlouw (TUD) *Modularization and Specification of Service-Oriented Systems*

22 Junte Zhang (UvA) *System Evaluation of Archival Description and Access*

23 Wouter Weerkamp (UvA) *Finding People and their Utterances in Social Media*

24 Herwin van Welbergen (UT) *Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior*

25 Syed Waqar ul Qounain Jaffry (VUA) *Analysis and Validation of Models for Trust Dynamics*

26 Matthijs Aart Pontier (VUA) *Virtual Agents for Human Communication: Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots*

27 Aniel Bhulai (VUA) *Dynamic website optimization through autonomous management of design patterns*

28 Rianne Kaptein (UvA) *Effective Focused Retrieval by Exploiting Query Context and Document Structure*

29 Faisal Kamiran (TUe) *Discrimination-aware Classification*

30 Egon van den Broek (UT) *Affective Signal Processing (ASP): Unraveling the mystery of emotions*

31 Ludo Waltman (EUR) *Computational and Game-Theoretic Approaches for Modeling Bounded Rationality*

32 Nees-Jan van Eck (EUR) *Methodological Advances in Bibliometric Mapping of Science*

33 Tom van der Weide (UU) *Arguing to Motivate Decisions*

34 Paolo Turrini (UU) *Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations*

35 Maaike Harbers (UU) *Explaining Agent Behavior in Virtual Training*

36 Erik van der Spek (UU) *Experiments in serious game design: a cognitive approach*

37 Adriana Burlutiu (RUN) *Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference*

38 Nyree Lemmens (UM) *Bee-inspired Distributed Optimization*

39 Joost Westra (UU) *Organizing Adaptation using Agents in Serious Games*

40 Viktor Clerc (VUA) *Architectural Knowledge Management in Global Software Development*

41 Luan Ibraimi (UT) *Cryptographically Enforced Distributed Data Access Control*

42 Michal Sindlar (UU) *Explaining Behavior through Mental State Attribution*

43 Henk van der Schuur (UU) *Process Improvement through Software Operation Knowledge*

44 Boris Reuderink (UT) *Robust Brain-Computer Interfaces*

45 Herman Stehouwer (UvT) *Statistical Language Models for Alternative Sequence Selection*

46 Beibei Hu (TUD) *Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work*

47 Azizi Bin Ab Aziz (VUA) *Exploring Computational Models for Intelligent Support of Persons with Depression*

48 Mark Ter Maat (UT) *Response Selection and Turn-taking for a Sensitive Artificial Listening Agent*

49 Andreea Niculescu (UT) *Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality*

2012

1 Terry Kakeeto (UvT) *Relationship Marketing for SMEs in Uganda*

2 Muhammad Umair (VUA) *Adaptivity, emotion, and Rationality in Human and Ambient Agent Models*

3 Adam Vanya (VUA) *Supporting Architecture Evolution by Mining Software Repositories*

4 Jurriaan Souer (UU) *Development of Content Management System-based Web Applications*

5 Marijn Plomp (UU) *Maturing Interorganisational Information Systems*

6 Wolfgang Reinhardt (OU) *Awareness Support for Knowledge Workers in Research Networks*

7 Rianne van Lambalgen (VUA) *When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions*

8 Gerben de Vries (UvA) *Kernel Methods for Vessel Trajectories*

9 Ricardo Neisse (UT) *Trust and Privacy Management Support for Context-Aware Service Platforms*

10 David Smits (TUe) *Towards a Generic Distributed Adaptive Hypermedia Environment*

11 J. C. B. Rantham Prabhakara (TUe) *Process Mining in the Large: Preprocessing, Discovery, and Diagnostics*

12 Kees van der Sluijs (TUe) *Model Driven Design and Data Integration in Semantic Web Information Systems*

13 Suleman Shahid (UvT) *Fun and Face: Exploring non-verbal expressions of emotion during playful interactions*

14 Evgeny Knutov (TUe) *Generic Adaptation Framework for Unifying Adaptive Web-based Systems*

15 Natalie van der Wal (VUA) *Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes*

16 Fiemke Both (VUA) *Helping people by understanding them: Ambient Agents supporting task execution and depression treatment*

17 Amal Elgammal (UvT) *Towards a Comprehensive Framework for Business Process Compliance*

18 Eltjo Poort (VUA) *Improving Solution Architecting Practices*

19 Helen Schonenberg (TUe) *What's Next? Operational Support for Business Process Execution*

20 Ali Bahramisharif (RUN) *Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing*

21 Roberto Cornacchia (TUD) *Querying Sparse Matrices for Information Retrieval*

22 Thijs Vis (UvT) *Intelligence, politie en veiligheidsdienst: verenigbare grootheden?*

23 Christian Muehl (UT) *Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction*

24 Laurens van der Werff (UT) *Evaluation of Noisy Transcripts for Spoken Document Retrieval*

25 Silja Eckartz (UT) *Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application*

26 Emile de Maat (UvA) *Making Sense of Legal Text*

27 Hayrettin Gurkok (UT) *Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games*

28 Nancy Pascall (UvT) *Engendering Technology Empowering Women*

29 Almer Tigelaar (UT) *Peer-to-Peer Information Retrieval*

30 Alina Pommeranz (TUD) *Designing Human-Centered Systems for Reflective Decision Making*

31 Emily Bagarukayo (RUN) *A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure*

32 Wietske Visser (TUD) *Qualitative multi-criteria preference representation and reasoning*

33 Rory Sie (OUN) *Coalitions in Cooperation Networks (COCOON)*

34 Pavol Jancura (RUN) *Evolutionary analysis in PPI networks and applications*

35 Evert Haasdijk (VUA) *Never Too Old To Learn: On-line Evolution of Controllers in Swarm- and Modular Robotics*

36 Denis Ssebugwawo (RUN) *Analysis and Evaluation of Collaborative Modeling Processes*

37 Agnes Nakakawa (RUN) *A Collaboration Process for Enterprise Architecture Creation*

38 Selmar Smit (VUA) *Parameter Tuning and Scientific Testing in Evolutionary Algorithms*

39 Hassan Fatemi (UT) *Risk-aware design of value and coordination networks*

40 Agus Gunawan (UvT) *Information Access for SMEs in Indonesia*

41 Sebastian Kelle (OU) *Game Design Patterns for Learning*

42 Dominique Verpoorten (OU) *Reflection Amplifiers in self-regulated Learning*

43 Anna Tordai (VUA) *On Combining Alignment Techniques*

44 Benedikt Kratz (UvT) *A Model and Language for Business-aware Transactions*

45 Simon Carter (UvA) *Exploration and Exploitation of Multilingual Data for Statistical Machine Translation*

46 Manos Tsagkias (UvA) *Mining Social Media: Tracking Content and Predicting Behavior*

47 Jorn Bakker (TUe) *Handling Abrupt Changes in Evolving Time-series Data*

48 Michael Kaisers (UM) *Learning against Learning: Evolutionary dynamics of reinforcement learning algorithms in strategic interactions*

49 Steven van Kervel (TUD) *Ontologogy driven Enterprise Information Systems Engineering*

50 Jeroen de Jong (TUD) *Heuristics in Dynamic Sceduling: a practical framework with a case study in elevator dispatching*

2013

1 Viorel Milea (EUR) *News Analytics for Financial Decision Support*

2 Erietta Liarou (CWI) *MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing*

3 Szymon Klarman (VUA) *Reasoning with Contexts in Description Logics*

4 Chetan Yadati (TUD) *Coordinating autonomous planning and scheduling*

5 Dulce Pumareja (UT) *Groupware Requirements Evolutions Patterns*

6 Romulo Goncalves (CWI) *The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience*

7 Giel van Lankveld (UvT) *Quantifying Individual Player Differences*

8 Robbert-Jan Merk (VUA) *Making enemies: cognitive modeling for opponent agents in fighter pilot simulators*

9 Fabio Gori (RUN) *Metagenomic Data Analysis: Computational Methods and Applications*

10 Jeewanie Jayasinghe Arachchige (UvT) *A Unified Modeling Framework for Service Design*

11 Evangelos Pournaras (TUD) *Multi-level Reconfigurable Self-organization in Overlay Services*

12 Marian Razavian (VUA) *Knowledge-driven Migration to Services*

13 Mohammad Safiri (UT) *Service Tailoring: User-centric creation of integrated IT-based homecare services to support independent living of elderly*

14 Jafar Tanha (UvA) *Ensemble Approaches to Semi-Supervised Learning Learning*

15 Daniel Hennes (UM) *Multiagent Learning: Dynamic Games and Applications*

16 Eric Kok (UU) *Exploring the practical benefits of argumentation in multi-agent deliberation*

17 Koen Kok (VUA) *The PowerMatcher: Smart Coordination for the Smart Electricity Grid*

18 Jeroen Janssens (UvT) *Outlier Selection and One-Class Classification*

19 Renze Steenhuizen (TUD) *Coordinated Multi-Agent Planning and Scheduling*

20 Katja Hofmann (UvA) *Fast and Reliable Online Learning to Rank for Information Retrieval*

21 Sander Wubben (UvT) *Text-to-text generation by monolingual machine translation*

22 Tom Claassen (RUN) *Causal Discovery and Logic*

23 Patricio de Alencar Silva (UvT) *Value Activity Monitoring*

24 Haitham Bou Ammar (UM) *Automated Transfer in Reinforcement Learning*

25 Agnieszka Anna Latoszek-Berendsen (UM) *Intention-based Decision Support. A new way of representing and implementing clinical guidelines in a Decision Support System*

26 Alireza Zarghami (UT) *Architectural Support for Dynamic Homecare Service Provisioning*

27 Mohammad Huq (UT) *Inference-based Framework Managing Data Provenance*

28 Frans van der Sluis (UT) *When Complexity becomes Interesting: An Inquiry into the Information eXperience*

29 Iwan de Kok (UT) *Listening Heads*

30 Joyce Nakatumba (TUe) *Resource-Aware Business Process Management: Analysis and Support*

31 Dinh Khoa Nguyen (UvT) *Blueprint Model and Language for Engineering Cloud Applications*

32 Kamakshi Rajagopal (OUN) *Networking For Learning: The role of Networking in a Lifelong Learner's Professional Development*

33 Qi Gao (TUD) *User Modeling and Personalization in the Microblogging Sphere*

34 Kien Tjin-Kam-Jet (UT) *Distributed Deep Web Search*

35 Abdallah El Ali (UvA) *Minimal Mobile Human Computer Interaction*

36 Than Lam Hoang (TUe) *Pattern Mining in Data Streams*

37 Dirk Börner (OUN) *Ambient Learning Displays*

38 Eelco den Heijer (VUA) *Autonomous Evolutionary Art*

39 Joop de Jong (TUD) *A Method for Enterprise Ontology based Design of Enterprise Information Systems*

40 Pim Nijssen (UM) *Monte-Carlo Tree Search for Multi-Player Games*

41 Jochem Liem (UvA) *Supporting the Conceptual Modelling of Dynamic Systems: A Knowledge Engineering Perspective on Qualitative Reasoning*

42 Léon Planken (TUD) *Algorithms for Simple Temporal Reasoning*

43 Marc Bron (UvA) *Exploration and Contextualization through Interaction and Concepts*

2014

1 Nicola Barile (UU) *Studies in Learning Monotone Models from Data*

2 Fiona Tuliyano (RUN) *Combining System Dynamics with a Domain Modeling Method*

3 Sergio Raul Duarte Torres (UT) *Information Retrieval for Children: Search Behavior and Solutions*

4 Hanna Jochmann-Mannak (UT) *Websites for children: search strategies and interface design - Three studies on children's search performance and evaluation*

5 Jurriaan van Reijsen (UU) *Knowledge Perspectives on Advancing Dynamic Capability*

6 Damian Tamburri (VUA) *Supporting Networked Software Development*

7 Arya Adriansyah (TUe) *Aligning Observed and Modeled Behavior*

8 Samur Araujo (TUD) *Data Integration over Distributed and Heterogeneous Data Endpoints*

9 Philip Jackson (UvT) *Toward Human-Level Artificial Intelligence: Representation and Computation of Meaning in Natural Language*

10 Ivan Salvador Razo Zapata (VUA) *Service Value Networks*

11 Janneke van der Zwaan (TUD) *An Empathic Virtual Buddy for Social Support*

12 Willem van Willigen (VUA) *Look Ma, No Hands: Aspects of Autonomous Vehicle Control*

13 Arlette van Wissen (VUA) *Agent-Based Support for Behavior Change: Models and Applications in Health and Safety Domains*

14 Yangyang Shi (TUD) *Language Models With Meta-information*

15 Natalya Mogles (VUA) *Agent-Based Analysis and Support of Human Functioning in Complex Socio-Technical Systems: Applications in Safety and Healthcare*

16 Krystyna Milian (VUA) *Supporting trial recruitment and design by automatically interpreting eligibility criteria*

17 Kathrin Dentler (VUA) *Computing healthcare quality indicators automatically: Secondary Use of Patient Data and Semantic Interoperability*

18 Mattijs Ghijsen (UvA) *Methods and Models for the Design and Study of Dynamic Agent Organizations*

19 Vinicius Ramos (TUe) *Adaptive Hypermedia Courses: Qualitative and Quantitative Evaluation and Tool Support*

20 Mena Habib (UT) *Named Entity Extraction and Disambiguation for Informal Text: The Missing Link*

21 Kassidy Clark (TUD) *Negotiation and Monitoring in Open Environments*

22 Marieke Peeters (UU) *Personalized Educational Games: Developing agent-supported scenario-based training*

23 Eleftherios Sidirourgos (UvA/CWI) *Space Efficient Indexes for the Big Data Era*

24 Davide Ceolin (VUA) *Trusting Semi-structured Web Data*

25 Martijn Lappenschaar (RUN) *New network models for the analysis of disease interaction*

26 Tim Baarslag (TUD) *What to Bid and When to Stop*

27 Rui Jorge Almeida (EUR) *Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty*

28 Anna Chmielowiec (VUA) *Decentralized k-Clique Matching*

29 Jaap Kabbedijk (UU) *Variability in Multi-Tenant Enterprise Software*

30 Peter de Cock (UvT) *Anticipating Criminal Behaviour*

31 Leo van Moergestel (UU) *Agent Technology in Agile Multiparallel Manufacturing and Product Support*

32 Naser Ayat (UvA) *On Entity Resolution in Probabilistic Data*

33 Tesfa Tegegne (RUN) *Service Discovery in eHealth*

34 Christina Manteli (VUA) *The Effect of Governance in Global Software Development: Analyzing Transactive Memory Systems*

35 Joost van Ooijen (UU) *Cognitive Agents in Virtual Worlds: A Middleware Design Approach*

36 Joos Buijs (TUe) *Flexible Evolutionary Algorithms for Mining Structured Process Models*

37 Maral Dadvar (UT) *Experts and Machines United Against Cyberbullying*

38 Danny Plass-Oude Bos (UT) *Making brain-computer interfaces better: improving usability through post-processing*

39 Jasmina Maric (UvT) *Web Communities, Immigration, and Social Capital*

40 Walter Omona (RUN) *A Framework for Knowledge Management Using ICT in Higher Education*

41 Frederic Hogenboom (EUR) *Automated Detection of Financial Events in News Text*

42 Carsten Eijckhof (CWI/TUD) *Contextual Multidimensional Relevance Models*

43 Kevin Vlaanderen (UU) *Supporting Process Improvement using Method Increments*

44 Paulien Meesters (UvT) *Intelligent Blauw: Intelligence-gestuurde politiezorg in gebiedsgebonden eenheden*

45 Birgit Schmitz (OUN) *Mobile Games for Learning: A Pattern-Based Approach*

46 Ke Tao (TUD) *Social Web Data Analytics: Relevance, Redundancy, Diversity*

47 Shangsong Liang (UvA) *Fusion and Diversification in Information Retrieval*

2015

1 Niels Netten (UvA) *Machine Learning for Relevance of Information in Crisis Response*

2 Faiza Bukhsh (UvT) *Smart auditing: Innovative Compliance Checking in Customs Controls*

3 Twan van Laarhoven (RUN) *Machine learning for network data*

4 Howard Spoelstra (OUN) *Collaborations in Open Learning Environments*

5 Christoph Bösch (UT) *Cryptographically Enforced Search Pattern Hiding*

6 Farideh Heidari (TUD) *Business Process Quality Computation: Computing Non-Functional Requirements to Improve Business Processes*

7 Maria-Hendrike Peetz (UvA) *Time-Aware Online Reputation Analysis*

8 Jie Jiang (TUD) *Organizational Compliance: An agent-based model for designing and evaluating organizational interactions*

9 Randy Klaassen (UT) *HCI Perspectives on Behavior Change Support Systems*

10 Henry Hermans (OUN) *OpenU: design of an integrated system to support lifelong learning*

11 Yongming Luo (TUe) *Designing algorithms for big graph datasets: A study of computing bisimulation and joins*

12 Julie M. Birkholz (VUA) *Modi Operandi of Social Network Dynamics: The Effect of Context on Scientific Collaboration Networks*

13 Giuseppe Procaccianti (VUA) *Energy-Efficient Software*

14 Bart van Straalen (UT) *A cognitive approach to modeling bad news conversations*

15 Klaas Andries de Graaf (VUA) *Ontology-based Software Architecture Documentation*

16 Changyun Wei (UT) *Cognitive Coordination for Cooperative Multi-Robot Teamwork*

17 André van Cleeff (UT) *Physical and Digital Security Mechanisms: Properties, Combinations and Trade-offs*

18 Holger Pirk (CWI) *Waste Not, Want Not!: Managing Relational Data in Asymmetric Memories*

19 Bernardo Tabuenca (OUN) *Ubiquitous Technology for Lifelong Learners*

20 Loïs Vanhée (UU) *Using Culture and Values to Support Flexible Coordination*

21 Sibren Fetter (OUN) *Using Peer-Support to Expand and Stabilize Online Learning*

22 Zhemin Zhu (UT) *Co-occurrence Rate Networks*

23 Luit Gazendam (VUA) *Cataloguer Support in Cultural Heritage*

24 Richard Berendsen (UvA) *Finding People, Papers, and Posts: Vertical Search Algorithms and Evaluation*

25 Steven Woudenberg (UU) *Bayesian Tools for Early Disease Detection*

26 Alexander Hogenboom (EUR) *Sentiment Analysis of Text Guided by Semantics and Structure*

27 Sándor Héman (CWI) *Updating compressed column-stores*

28 Janet Bagorogoza (TiU) *Knowledge Management and High Performance: The Uganda Financial Institutions Model for HPO*

29 Hendrik Baier (UM) *Monte-Carlo Tree Search Enhancements for One-Player and Two-Player Domains*

30 Kiavash Bahreini (OUN) *Real-time Multimodal Emotion Recognition in E-Learning*

31 Yakup Koç (TUD) *On Robustness of Power Grids*

32 Jerome Gard (UL) *Corporate Venture Management in SMEs*

33 Frederik Schadd (UM) *Ontology Mapping with Auxiliary Resources*

34 Victor de Graaff (UT) *Geosocial Recommender Systems*

35 Junchao Xu (TUD) *Affective Body Language of Humanoid Robots: Perception and Effects in Human Robot Interaction*

2016

1 Syed Saiden Abbas (RUN) *Recognition of Shapes by Humans and Machines*

2 Michiel Christiaan Meulendijk (UU) *Optimizing medication reviews through decision support: prescribing a better pill to swallow*

3 Maya Sappelli (RUN) *Knowledge Work in Context: User Centered Knowledge Worker Support*

4 Laurens Rietveld (VUA) *Publishing and Consuming Linked Data*

5 Evgeny Sherkhonov (UvA) *Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers*

6 Michel Wilson (TUD) *Robust scheduling in an uncertain environment*

7 Jeroen de Man (VUA) *Measuring and modeling negative emotions for virtual training*

8 Matje van de Camp (TiU) *A Link to the Past: Constructing Historical Social Networks from Unstructured Data*

9 Archana Nottamkandath (VUA) *Trusting Crowdsourced Information on Cultural Artefacts*

10 George Karafotias (VUA) *Parameter Control for Evolutionary Algorithms*

11 Anne Schuth (UvA) *Search Engines that Learn from Their Users*

12 Max Knobbout (UU) *Logics for Modelling and Verifying Normative Multi-Agent Systems*

13 Nana Baah Gyan (VU) *The Web, Speech Technologies and Rural Development in West Africa: An ICT4D Approach*

14 Ravi Khadka (UU) *Revisiting Legacy Software System Modernization*

15 Steffen Michels (RUN) *Hybrid Probabilistic Logics: Theoretical Aspects, Algorithms and Experiments*

16 Guangliang Li (UvA) *Socially Intelligent Autonomous Agents that Learn from Human Reward*

17 Berend Weel (VUA) *Towards Embodied Evolution of Robot Organisms*

18 Albert Meroño Peñuela (VUA) *Refining Statistical Data on the Web*

19 Julia Efremova (TUe) *Mining Social Structures from Genealogical Data*

20 Daan Odijk (UvA) *Context & Semantics in News & Web Search*

This thesis presents research towards a core aim of IR: providing users with easy access to information. Three research themes guide the research presented in this thesis, contributing to three aspects of IR research: the domain in which an IR system is used, the users interacting with the system, and the different access scenarios in which these users engage with an IR system. Central to these research themes is the aim to gain insights into the behavior of searchers and develop algorithms to support them in their quest, whether it is a researcher exploring or studying a large collection, a web searcher struggling to find something, or a television viewer searching for related content.

The first research theme is motivated by the information seeking tasks of researchers exploring and studying large collections. To enable their search on a larger scale, we propose computational methods to connect collections and to infer the perspective offered in a news story. Motivated by how historians select documents for close reading, we propose novel methods for connecting collections using automatically extracted temporal references. To illustrate how these algorithms can be used to automatically create connections between collections, we introduce a novel search interface to explore and analyze the connected collections. The interface highlights different perspectives and requires little domain knowledge. Based on how communication scientists study framing in news, we propose an automatic thematic content analysis approach.

The second research theme is addressed in a mixed-methods study on how web searchers behave when they cannot find what they are looking for. Based on large-scale log analysis, crowd-sourced labeling, and predictive modeling, we show behavioral differences given task success and failure. Based on these findings we propose ways in which systems can reduce struggling in search. To support searchers, we propose and evaluate algorithms that accurately predict the nature of future actions and their anticipated impact on search outcomes. Our findings have implications for the design of search systems that help searchers struggle less and succeed more.

In the third and final research theme, we consider a pro-active search scenario, specifically in a live television setting. We propose algorithms that leverage contextual information to retrieve diverse related content for a leaned-back TV viewer. While watching television, people increasingly consume additional content related to what they are watching. Two methods to automatically retrieve content based on subtitles are introduced, one using entity linking, and one that uses reinforcement learning to generate effective queries for finding related content. Both methods are highly efficient and are currently used in a live television setting in near real time.

Each research chapter in this thesis provides insights and algorithms that help searchers when using IR applications. For varying domains, users, and access scenarios, the research presented in this thesis improves the ease of access to information.

ISBN 978-94-6182-686-2

9 789461 826862 >